

Kandidatuppsats i Statistik  
732G47

# Prediktion för arbetslösheten på Åland med LSTM-GRU

Ett arbete där rekursiva neurala nätverk med tre osäkerheter i beaktande predikterar det relativa arbetslöshetstalet på Åland

Eric Sjölund  
Maximus Schempp



Avdelningen för Statistik och maskininlärning  
Institutionen för datavetenskap  
Linköpings universitet  
Vårterminen, 2024 | LIU-IDA/STAT-G-24/003-SE

Handledare: Isak Hietala, Universitetsadjunkt  
Examinator: Linda Wänström, Universitetslektor, Docent

## Sammanfattning

Arbetslöshet är ett centralt ämne inom världsekonomin och används som ett av flera instrument för att mäta ett lands ekonomiska hälsa. Arbetslösheten är en komplex utmaning för alla världens länder och den har en genomgripande inverkan för hela världens individer och samhällen. Att snabbt och smidigt kunna mäta arbetslösheten genom att få fram tillförlitliga prediktioner på hur arbetslöshetsnivåerna kommer att se ut i framtiden kan vara av stor nytta för beslutsfattande om vilka åtgärder som behöver tas i akt.

ÅSUB (Ålands statistik- och utredningsbyrå) stod som uppdragsgivare för arbetet och hade som önskan att låta undersöka ifall maskininlärningsmetoder lämpade sig för att prediktera arbetslösheten. Vidare stod även ÅSUB för all data som användes under arbetet.

På uppdragsgivarens önskan syftade just detta arbete till att undersöka lämpligheten i att låta ett neuralt nätverk med LSTM-GRU-celler prediktera arbetslösheten på Åland. Tidigare studier har innan kunnat påvisa att sådana metoder visat framgångsrika resultat vid prediktion av arbetslöshet.

Resultatet bestod av sex stycken modeller där tre av dem predikterade arbetslösheten för 2024 och tre av dem arbetslösheten för 2019, detta för att ta hänsyn till covid-19. För 2019 tog en modell hänsyn till epistemisk osäkerhet, en annan till aleatorisk osäkerhet, och den tredje till en kombination av båda. För 2024 användes samma uppdelning, så att varje typ av osäkerhet beaktades totalt två gånger, en gång för varje år.

Den bästa modellen visade sig vara den som predikterade arbetslösheten för år 2019 och som tog en epistemisk osäkerhet i beaktande där det relativa arbetslöshetstalet predikterades till 3.51 % med en osäkerhet på mellan 2.68 och 4.3 procentenheter. Då det presenterade relativa arbetslöshetstalet för år 2019 sattes till 3.52 % formulerades slutsatsen till att när världsomvälvande avvikelser ej sker, som till exempel covid-19 så är det lämpligt att låta en maskininlärningsmetod prediktera arbetslösheten.



## Abstract

Unemployment is a central topic in the world economy and is used as one of several instruments to measure a country's economic health. Unemployment is a complex challenge for all countries of the world and it has a pervasive impact on individuals and societies throughout the world. Being able to quickly and easily measure unemployment by producing reliable predictions of what unemployment levels will look like in the future can be of great benefit for decision-making about which measures need to be taken into account.

ÅSUB (Statistics and Research Åland) was the client for the work and wanted to investigate whether machine learning methods were suitable for predicting unemployment. Furthermore, ÅSUB was also responsible for all data used during the work.

At the client's request, this particular work aimed to investigate the suitability of letting a neural network with LSTM-GRU cells predict unemployment on Åland. Previous studies have been able to demonstrate that such methods have shown successful results in predicting unemployment.

The result consisted of six models where three of them predicted the unemployment for 2024 and three of them the unemployment for 2019, this to take care of covid-19. For 2019, one model accounted for epistemic uncertainty, another for aleatoric uncertainty, and the third for a combination of both. For 2024, the same breakdown was used, so that each type of uncertainty was considered twice in total, once for each year.

The best model turned out to be the one that predicted unemployment for the year 2019 and that took epistemic uncertainty into account where the relative unemployment rate was predicted at 3.51 % with an uncertainty of between 2.68 and 4.3 percentage points. When the presented relative unemployment figure for the year 2019 was set at 3.52 %, the conclusion was formulated that when world-changing deviations do not occur, such as covid-19, it is appropriate to let a machine learning method predict unemployment.



## **Förord**

Tack till ÅSUB som har tillhandahållit oss data för detta arbete. Vi vill även rikta ett stort tack till Isak Hietala som varit vår handledare under arbetet.





# Innehåll

<b>1</b>	<b>Introduktion</b>	<b>1</b>
1.1	Bakgrund . . . . .	1
1.2	Syfte . . . . .	2
1.3	Etiska och samhällsliga aspekter . . . . .	3
<b>2</b>	<b>Data</b>	<b>5</b>
2.1	Relativt arbetslöshetstal . . . . .	6
2.2	Ungdomsarbetslöshetstal . . . . .	7
2.3	Lediga platser / arbetslösa . . . . .	8
2.4	Konsumentprisindex . . . . .	9
2.5	Inresande till Åland . . . . .	10
2.6	Näringslivets löneutbetalningar . . . . .	11
2.7	Näringslivets omsättning . . . . .	13
2.8	Antal covid-19 fall . . . . .	15
<b>3</b>	<b>Metod</b>	<b>17</b>
3.1	Aktiveringsfunktioner . . . . .	19
3.1.1	Sigmoid . . . . .	19
3.1.2	Tanh . . . . .	20
3.2	Batch . . . . .	21
3.3	Backpropagation . . . . .	21
3.4	Optimering . . . . .	22
3.4.1	ADAM . . . . .	23
3.5	Regularisering . . . . .	23
3.5.1	Early stopping . . . . .	24
3.5.2	Dropout . . . . .	24
3.6	Osäkerheter . . . . .	25
3.6.1	Epistemisk osäkerhet . . . . .	25
3.6.2	Aleatorisk Osäkerhet . . . . .	27
3.6.3	Kombination epistemisk och aleatorisk osäkerhet . . . . .	27

3.7	Recurrent Neural Network . . . . .	27
3.7.1	Input för RNN . . . . .	28
3.7.2	Struktur RNN . . . . .	29
3.7.2.1	Backpropagation Through Time . . . . .	31
3.7.2.2	Problematik med gradient vanishing och gradient explosion . . . . .	31
3.7.2.3	Exempel: Backpropagation Through Time . . . . .	32
3.8	Long Short Term Memory . . . . .	32
3.9	Gated Recurrent Unit . . . . .	37
<b>4</b>	<b>Resultat</b>	<b>43</b>
4.1	Implementering i R och Python . . . . .	44
4.2	LSTM-GRU 1 . . . . .	47
4.3	LSTM-GRU 2 . . . . .	48
4.4	LSTM-GRU 3 . . . . .	49
4.5	LSTM-GRU 4 . . . . .	50
4.6	LSTM-GRU 5 . . . . .	51
4.7	LSTM-GRU 6 . . . . .	52
<b>5</b>	<b>Diskussion/Analys</b>	<b>53</b>
5.1	Prediktioner . . . . .	53
5.2	Studiens jämförbarhet . . . . .	54
5.3	Modellutvärdering . . . . .	54
5.4	Framtida forskning . . . . .	55
<b>6</b>	<b>Slutsatser</b>	<b>57</b>
	<b>Litteraturförteckning</b>	<b>61</b>

# Figurer

2.1	Relativt arbetslöshetstal(%), 2001-2023 . . . . .	6
2.2	Ungdomsarbetslöshetstal < 25år(%), 2001-2023 . . . . .	7
2.3	Lediga platser / Arbetslösa(%), 2001-2023 . . . . .	8
2.4	Konsumentprisindex(%), 2001-2023 . . . . .	9
2.5	Total antal inresande till Åland, 2001-2023 . . . . .	10
2.6	Näringslivets totala löneutbetalningar, 2001-2023 . . . . .	12
2.7	Näringslivets totala omsättning, 2001-2023 . . . . .	14
2.8	Antal covid-19 fall, 2001-2023 . . . . .	15
3.1	Neuralt nätverk . . . . .	17
3.2	Aktiveringsfunktionen Sigmoid . . . . .	20
3.3	Aktiveringsfunktionen tanh . . . . .	21
3.4	Early Stopping visualiserat . . . . .	24
3.5	Dropout visualiserat i ett neuralt nätverk . . . . .	25
3.6	MC-Dropout visualiserat . . . . .	26
3.7	Fönstring av en variabel . . . . .	28
3.8	RNN-struktur med ett gömt lager . . . . .	29
3.9	Utvecklad RNN . . . . .	30
3.10	Cellstruktur LSTM . . . . .	33
3.11	Forget gate LSTM . . . . .	34
3.12	Input gate LSTM . . . . .	35
3.13	Output gate LSTM . . . . .	36
3.14	Cellstruktur GRU . . . . .	38
3.15	Reset gate GRU . . . . .	39
3.16	Update gate GRU . . . . .	40
4.1	Relativt arbetslöshetstal(%), 2001-2023 . . . . .	43
4.2	Prediktion av Relativt arbetslöshetstal 2024 för respektive modell . . . . .	45
4.3	Prediktion av Relativt arbetslöshetstal 2019 för respektive modell . . . . .	46
4.4	LSTM-GRU med epistemisk osäkerhet . . . . .	47

4.5 LSTM-GRU med aleatorisk osäkerhet . . . . .	48
4.6 LSTM-GRU med epistemisk och aleatorisk osäkerhet . . . . .	49
4.7 LSTM-GRU med epistemisk osäkerhet, beskuret datamaterial . . . . .	50
4.8 LSTM-GRU med aleatorisk osäkerhet, beskuret datamaterial . . . . .	51
4.9 LSTM-GRU med epistemisk och aleatorisk osäkerhet, beskuret datamaterial . . . . .	52

# Tabeller

4.1 Modellinformation . . . . . 44



# 1. Introduktion

Uppdragsgivaren till detta arbete är ÅSUB (Ålands statistik- och utredningsbyrå). ÅSUB är en oberoende institution vars ansvar är att bedriva forskning och utredningsverksamhet men därtill även producera och publicera officiell statistik om, och från Åland. Deras centrala område för forskningsverksamheten ligger i det samhällsekonomiska men forskningen omfattar även 23 andra samhällsområden. ÅSUB producerar officiell statistik för debatter, beslutsfattande samt forskning. I ÅSUB:s statistiska årsbok för 2023 nämner de bland annat de krav som ställs på statistik är dess relevans, jämförbarhet samt att den ska vara korrekt och oberoende. Vidare nämner de att en viktig byggsten i varje demokrati är tillgången till lättåtkomlig samt tillförlitlig statistik. Enligt Statens offentliga utredningar(SOU) avses officiell statistik som den statistiken som publiceras av en utsedd ansvarig myndighet som i regel ska vara objektiv, lättillgänglig och finnas för utredningsverksamhet, forskningssyften eller för allmän information [1]. I Sverige är denna ansvariga myndighet Statistiska centralbyrån (SCB).

## 1.1 Bakgrund

Arbetslöshet är ett centralt ämne inom världsekonomin och en komplex utmaning med inverkan på hela världens olika samhällen. Arbetslösheten fungerar som ett instrument för att mäta ett lands ekonomiska hälsa och kan medföra genomgripande följder för dels individer men därtill även samhällen. [2]. Att skapa sig en förståelse för arbetslösheten genom att prediktera dess mönster i en tidsserie kan ha en stor innebörd och vara till hjälp för att skapa insikter om beslut till politiska åtgärder för att gagna ekonomisk stabilitet. [3]

För att analysera och prediktera arbetslösheten för en längre period framöver eller inom det närmast kommande året har ofta traditionella metoder använts som byggts på historiska data och renodlade statistiska modeller, i takt med att tekniken utvecklats har dock maskininlärningsmetoder börjat introducerats mer inom statistiken.

Flera studier har kunnat påvisa att maskininlärningsmetoder är ett framgångsrikt sätt att förutsäga arbetslösheten [4] [5] [6]. Det har även påvisats att maskininlärningsmodeller som neurala nätverk med en hybrid av LSTM (Long Short Term Memory) och GRU (Gated Recurrent Unit) [3] har kunnat användas framgångsrikt för att prediktera arbetslösheten i ett land.

Mustafa Yurtsever [3] publicerade i juni 2023 en artikel där han med hjälp av maskininläring predikterade arbetslösheten i USA, UK, Frankrike och Italien. Han skapade tre neurala nätverk med LSTM, GRU och en kombination av dem med syftet att fastställa vilken av dessa tre nätverk som genererar den minsta osäkerheten i skattningarna med hjälp av måtten RMSE (Root Mean Squared Error) och MAE (Mean Absolute Error) och således dra en slutsats om vilken nätverksmodell som ger mest träffsäkra prediktioner på arbetslöshet. Data bestod av totalt 473 månader från januari 1983 till maj 2022. Han delade upp data så att 70 % blev en träningsmängd och 30 % en testmängd. Resultatet gav prediktioner som alla i stort sett gav samma värde som den sanna arbetslöshetsnivån för varje land med en viss avvikelse för USA där den predikterade

arbetslösheten vid vissa tidpunkter tenderade att prediktera högre arbetslöshet än vad som var uppmätt. Vidare visade det sig att för alla länder utom Italien så var en kombination av LSTM och GRU den metod som genererade lägst osäkerhet sett till RMSE och MAE. Yurtsever nämner i slutsatsen av sitt arbete att möjliga begränsningar med modellen kan komma att uppstå vid prediktion av andra länders arbetslöshet, eller vid olika tidpunkter, då resultaten kan variera från de han fick på grund av enskilda sociala och ekonomiska faktorer som speglar arbetslöshetsnivåerna där.

Vad det gäller de mer traditionsenliga metoderna har studier också kommit fram till slutsatsen att ARIMAX (Auto Regressive Intergrated Moving Average with Exogenous variable) skapat prognoser som varit tillförlitliga och nära verkligheten, där man bland annat påpekat att de exogena variablerna som läggs till i modellen kan fånga upp händelser som händer i nuläget om data funnits tillgängligt i tiden då man arbetat med prognoserna. [1] I november 2020 publicerade Muhammad Fajar et al. [6] en artikel där de med en ARIMAX-modell syftade till att prediktera arbetslösheten under covid-19-pandemin i Indonesien. På grund av bristfälliga data använde de sig av interpolering för att vid modelleringen ha en datamängd på 186 observationer från januari 2004 till juni 2020 där de sedan delade upp data så att januari 2004 till januari 2020 användes som träningsmängd och februari 2020 till juni 2020 som testmängd, vidare använde de även data över antalet permitteringar eller uppsägningar över samma period som exogen variabel. Resultatet genererade en prognos där deras uppskattade arbetslöshet för februari 2020 låg på 5.46 %, ungefär 0.5 procentenheter ifrån den faktiska arbetslösheten som låg på 4.99 % i februari 2020. I diskussionen nämner de att ARIMAX-modellen inför framtiden kan integreras med delar av en markovmodell för att även kunna erhålla skattningarnas sannolikhet.

## 1.2 Syfte

Detta arbete kommer att undersöka potentialen hos en metod baserad på maskinlärning. Modellen som kommer att användas och undersökas är en maskininlärningsbaserad modell i form av ett neuralt nätverk med en hybrid av LSTM och GRU. I arbetet kommer totalt 34 variabler från åtta makroekonomiska källor att hämtas in, kombineras och analyseras med strävan efter att komma fram till så precisa prediktioner på Ålands arbetslöshet under 2024 som möjligt.

Därav kan syftet med uppsatsen sammanfattas som en undersökning i hur lämpligt det är att skapa prognoser med hjälp av maskininlärningsmetoder och en frågeställning kan formuleras enligt följande:

Är det lämpligt att använda sig av maskininlärning vid prognos av arbetslöshet?



### 1.3 Etiska och samhällsliga aspekter

Data har hämtats från ÅSUB:s statistikdatabas med officiell statistik, data har bearbetats enligt GDPR och Statistiklagen på Åland så att uppgifter om en enskild individ inte ska kunna röjas.[7]

Maskininlärning används idag inom flera olika områden för att analysera data. Med maskininlärning öppnas möjligheterna för automatiserade metoder som dels bidrar till nya insikter, träffsäkrare analyser och snabbare tillvägagångssätt för att bland annat prediktera arbetslöshet eller annan typ av data på samhällslig nivå. Genom att kunna producera mer precisa prognoser kan samhället ha en vinning ur det sett från en förberedande aspekt, då myndigheter på förhand kan använda prediktionerna till beslutsfattande om vilka åtgärder som kommer att behöva vidtas för att smidigt kunna hantera ett stundande läge. Sett ur ett privat perspektiv har även företag stor vinning i modeller som är bättre på att prediktera framtiden för att kunna ta hänsyn till framtida intäkter kontra utgifter, mäta lönsamhet över tid och väl där använda fördelarna för beslutsfattning kring nästkommande perioder. Andra fördelar med att ha modeller som bättre predikterar framtiden är att samhället även kommer närmare sanningen, vilket leds in på det som togs upp ovan, nämligen att det lägger en stabilare grund för framtida beslutsfattning men även för vidare forskning. [8] Vidare är vår övertygelse att ytterligare en gren inom statistiken bidrar till ett bredare utbud om hur olika processer för analys av data kan tillämpas och således medför mer precisa resultat in i framtiden.



## 2. Data

Här ges en beskrivning av var data kommer ifrån. Här ges också en kortfattad beskrivning av variabler och observationer i det givna datamaterialet samt av eventuella transformationer och skapande av nya variabler. Data kommer huvudsakligen från ÅSUBs statistikdatabas [9] med avvikelse för data för covid-19 som är hämtad från institutet för hälsa och välfärd (på finska Terveystiete ja hyvinvoinnin laitos (THL)) [10] . Om inget annat anges så avser samtliga datamängder perioden januari 2001 till och med december 2023.

## 2.1 Relativt arbetslöshetstal

Relativt arbetslöshetstal är en datamängd av tidsseriekaraktär hämtad från ÅSUBs statistikdatabas [11]. Tidsserien beskriver hur den relativa arbetslösheten fluktuerat månadsvis genom åren. Data har bearbetats och beskrivits så att endast aktuell variabel och tidsperiod används. De som klassificeras som arbetslösa arbetssökande, per ÅSUBs definition [12], är de individer som för närvarande inte innehar någon anställning, inte driver företagsverksamhet på heltid och inte studerar på heltid. Elever och studenter på heltid betraktas inte heller som arbetslösa under ledigheter från skolan. Dessutom inkluderas de som har en anställning men är permitterade på heltid, samt de vars ordinarie arbetstid per vecka är mindre än fyra timmar i definitionen av arbetslöshet. Relativt arbetslöshetstal utger den procentandel av arbetskraften som befinner sig mellan åldrarna 15 till 74 år och som är arbetslösa. Arbetskraften beskrivs i sin tur som **sysselsatta + arbetslösa**. Relativa arbetslöshetstalet för befolkningen beräknas sedan som den procentandel arbetslösa som är i samma ålder i arbetskraften.



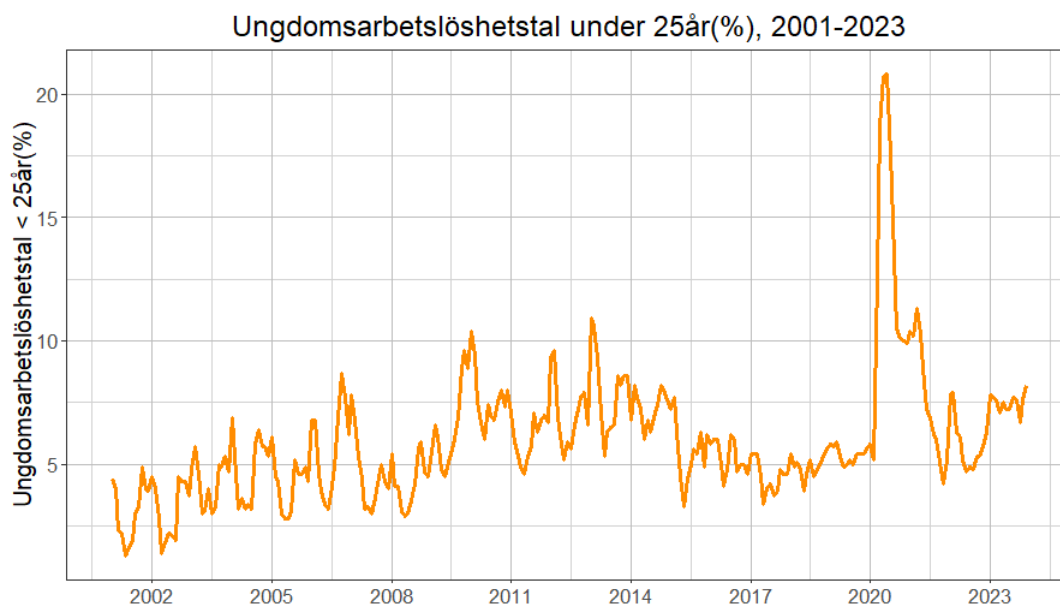
Källa: Ålands statistik- och utredningsbyrå (ÅSUB)

Figur 2.1: Relativt arbetslöshetstal(%), 2001-2023

I figuren syns det att det förekommer en stor spik vid 2020, då covid-19-pandemin inträffade. Utöver detta verkar det finnas en positiv trend för relativa arbetslösheten. Det verkar även finnas ett säsongsmönster med en spik vid januari från 2001-2015. Efter 2015 förekommer två spikar, en i januari och en i juli.

## 2.2 Ungdomsarbetslöshetstal

Ungdomsarbetslöshetstalet är en datamängd av tidsseriekaraktär hämtad från ÅSUBs statistikdatabas [11]. Tidsserien beskriver hur ungdomsarbetslösheten fluktuerat månadsvis genom åren. Data har bearbetats och beskurits så att endast aktuell variabel och tidsperiod används. Till ungdomsarbetslösa räknas den del av befolkningen som befinner sig i arbetskraften och är under 25 år.



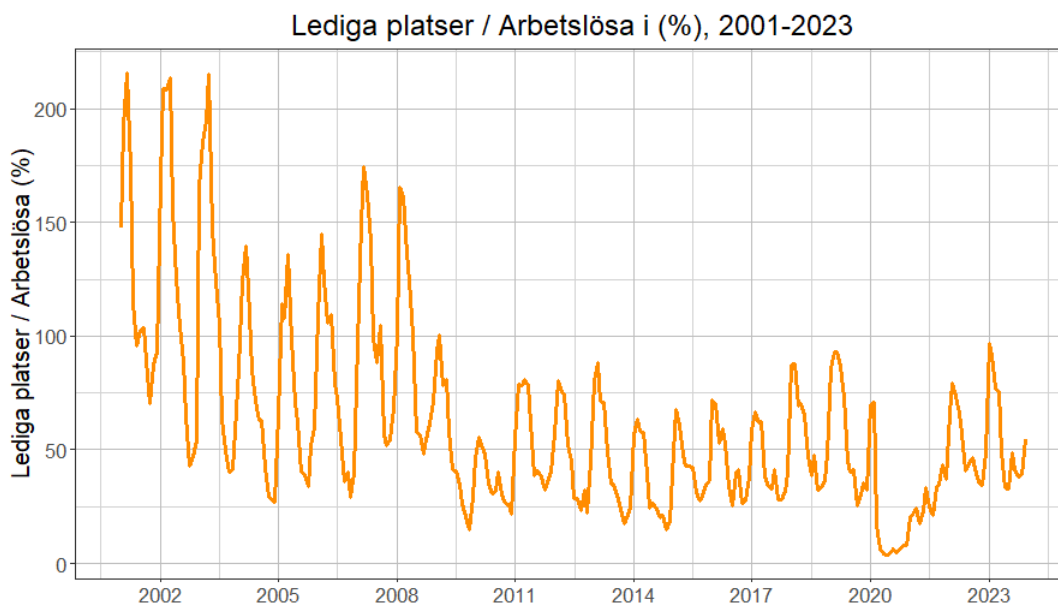
Källa: Ålands statistik- och utredningsbyrå (ÅSUB)

Figur 2.2: Ungdomsarbetslöshetstal < 25år(%), 2001-2023

I figur 2.2 syns det att jämfört med figur 2.1 verkar ungdomsarbetslösheten fluktuera något mer. Även under covid-19-pandemin är ungdomsarbetslöshetstalet avsevärt högre med en topp på över 20%.

## 2.3 Lediga platser / arbetslösa

Lediga platser / arbetslösa är en datamängd av tidsseriekaraktär hämtad från ÅSUBs statistikdatabas [11]. Tidsserien beskriver hur förhållandet mellan lediga platser gentemot arbetslösa fluktuerat månadsvis genom åren. Data har bearbetats och beskrivits så att endast aktuell variabel och tidsperiod används. Lediga jobb beskriver samtliga yrken som varit lediga under den aktuella månaden. Observera att det även kan vara jobb som var lediga föregående månad samt nyregistrerade lediga jobb.



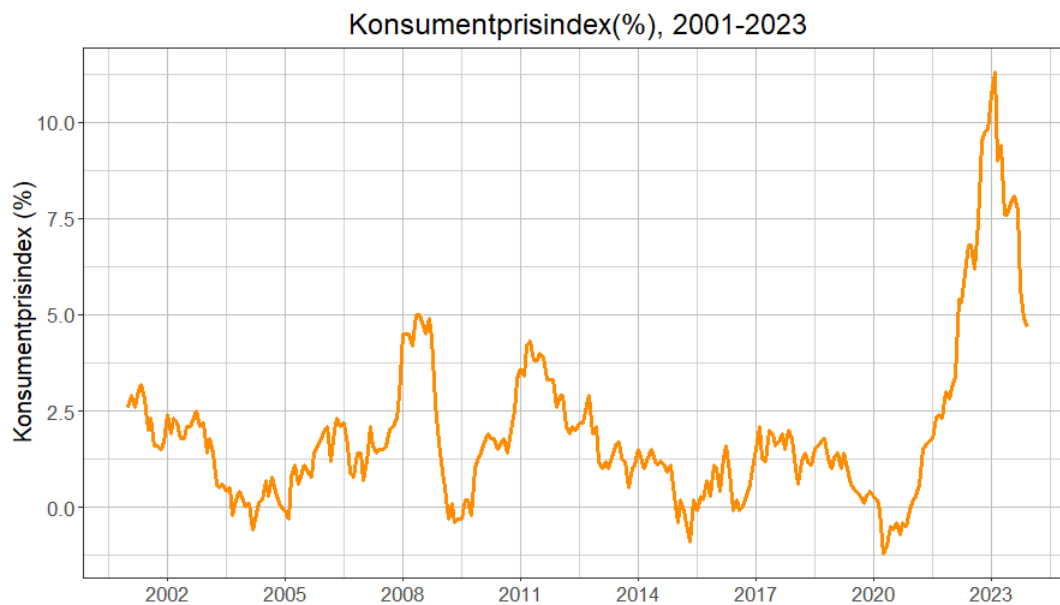
Källa: Ålands statistik- och utredningsbyrå (ÅSUB)

Figur 2.3: Lediga platser / Arbetslösa(%), 2001-2023

I figur 2.3 syns det att Lediga platser / arbetslösa har sjunkit från 2001-2023. Under 2020-2021 och till viss del under 2022 var denna kvot även låg men verkar återhämtat sig till det normala säsongsmonstret som går att observera från 2011-2019

## 2.4 Konsumentprisindex

Konsumentprisindex (KPI) är en datamängd av tidsseriekaraktär hämtad från ÅSUBs statistikdatabas [13]. Tidsserien beskriver hur KPI fluktuerat månadsvis genom åren. Data har bearbetats och beskrivits så att endast aktuell variabel och tidsperiod används. KPI beskriver ett index baserat på hur priset för ett antal olika varor och tjänster utvecklas över tid. Varor och tjänster som mäts kan tillkännas av livsmedel, alkohol och tobak, skor, transporter, kläder och annat. Därefter sammanställs priserna med ett vägningstal med syfte att avspegla den privata konsumtionen vid aktuell tid så långt som möjligt. KPI utgår alltid från ett basår.



Källa: Ålands statistik- och utredningsbyrå (ÅSUB)

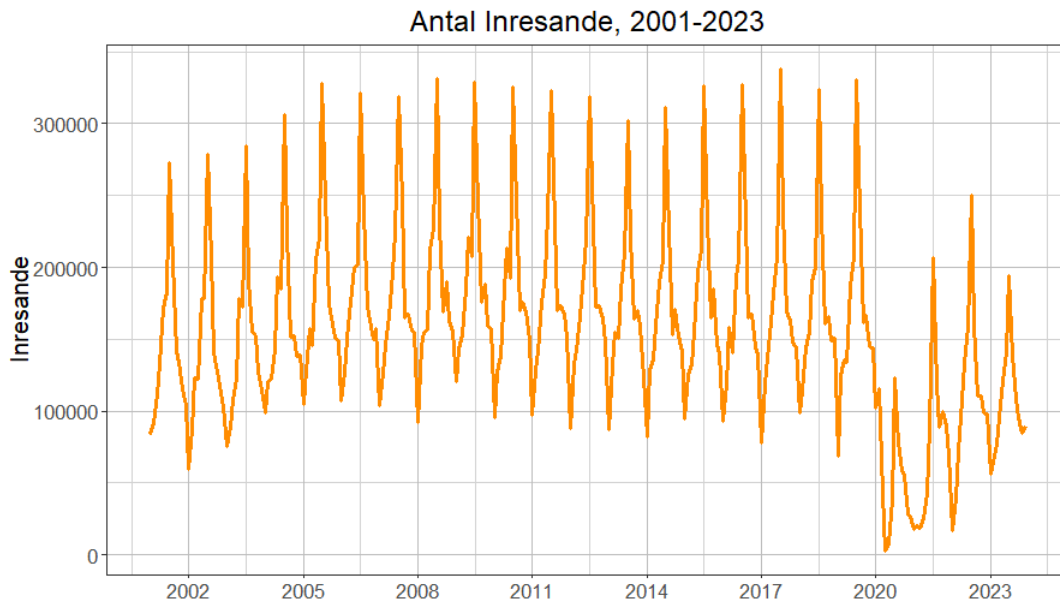
Figur 2.4: Konsumentprisindex(%), 2001-2023

I figur 2.4 syns det att KPI genom åren legat på en nivå mellan 0 till 2.5 % men haft två toppar vid 2008 och 2011. 2020 syns den lägst mätta nivån av KPI innan det sticker iväg och en spik förekommer på grund av covid-19. Från 2023 har serien börjat avta och ligger vid senaste mätningen på samma nivå som vid 2008.

## 2.5 Inresande till Åland

Inresande till Åland är en datamängd av tidsseriekaraktär hämtad från ÅSUBs statistikdatabas [14]. Tidsserien beskriver det totala inresandet till Åland genom åren. Data har bearbetats och beskrivits så att endast aktuell variabel och tidsperiod används. Inresande till Åland utgörs av följande variabler:

- Totalt
- Totalt från Finland
- Från Finland med skärgårdsfärjor
- Totalt från Sverige
- Från Sverige med kryssningsfartyg
- Totalt från övriga länder



Källa: Ålands statistik- och utredningsbyrå (ÅSUB)

Figur 2.5: Totalt antal inresande till Åland, 2001-2023

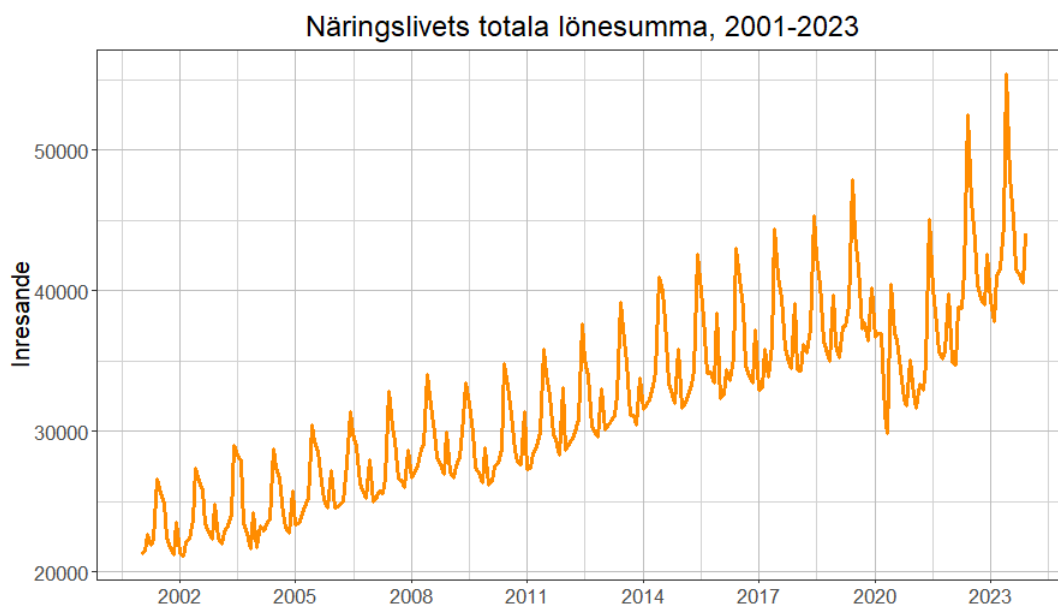
I figur 2.5 syns nu det totala inresandet till Åland. Det går att observera ett tydligt säsongsmönster där inresande till Åland verkar toppa vid juli varje år, annars är serien relativt jämn genom åren men avtar kraftigt år 2020 på grund av covid-19 men hämtar sig sedan sakta tillbaka till tidigare nivåer.



## 2.6 Näringslivets löneutbetalningar

Näringslivets löneutbetalningar är en datamängd av tidsseriekaraktär hämtad från ÅSUBs statistikdatabas [15]. Tidsserien beskriver hur totala löneutbetalningarna för elva olika branscher fluktuerat genom åren. Data har bearbetats och beskrivits så att endast aktuella variabler och aktuell tidsperiod används. Näringslivet utgörs här av tolv variabler bestående av de elva branscherna:

- Primärnäringsar
- Livsmedelsindustri
- Övrig industri
- Vatten-, gas-, el-, värme- och avloppshantering
- Byggverksamhet
- Handel
- Transport
- Hotell och restaurang
- Finans och försäkring
- Företagstjänster
- Personliga tjänster
- Totalsumma



Källa: Ålands statistik- och utredningsbyrå (ÅSUB)

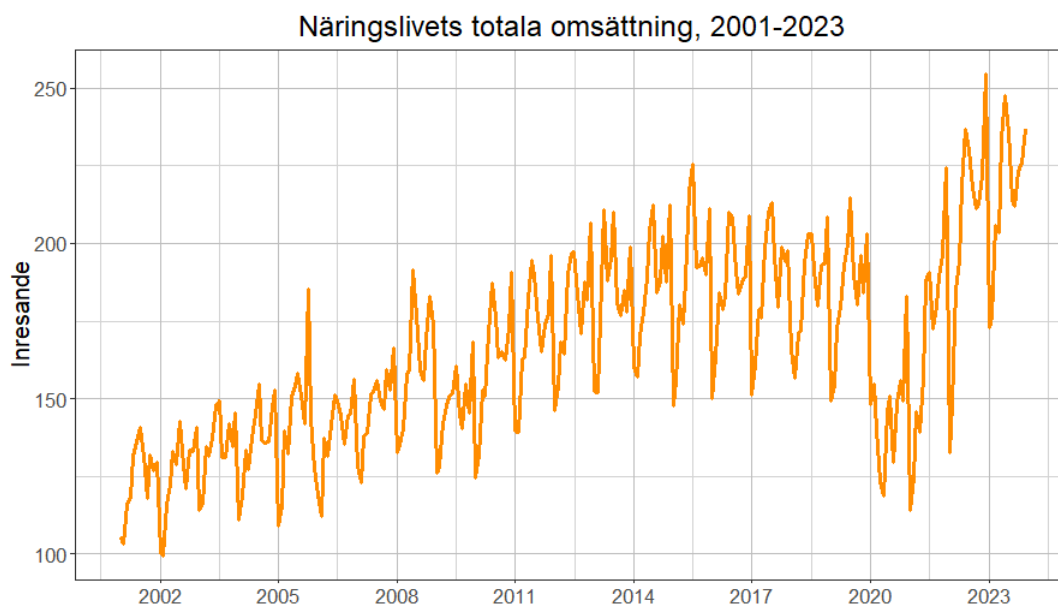
Figur 2.6: Näringslivets totala löneutbetalningar, 2001-2023

I figur 2.6 syns det att de totala lönesummorna successivt ökat genom åren, med en viss avvikelse för covid-19 vid 2020, men som sedan hämtat sig och vid 2023 fortsatt verkar öka i nivå, det syns att lönesummorna är som högst i slutet av 2022.

## 2.7 Näringslivets omsättning

Näringslivets omsättning är en datamängd av tidsseriekaraktär hämtad från ÅSUBs statistikdatabas [16]. Tidsserien beskriver hur den totala omsättningen för elva olika branscher fluktuerat genom åren. Data har bearbetats och beskrivits så att endast aktuella variabler och aktuell tidsperiod används. Näringslivets utgörs av tolv variabler bestående av de elva branscherna:

- Primärnäringsar
- Livsmedelsindustri
- Övrig industri
- Vatten-, gas-, el-, värme- och avloppshantering
- Byggverksamhet
- Handel
- Transport
- Hotell och restaurang
- Finans och försäkring
- Företagstjänster
- Personliga tjänster
- Totalsumma



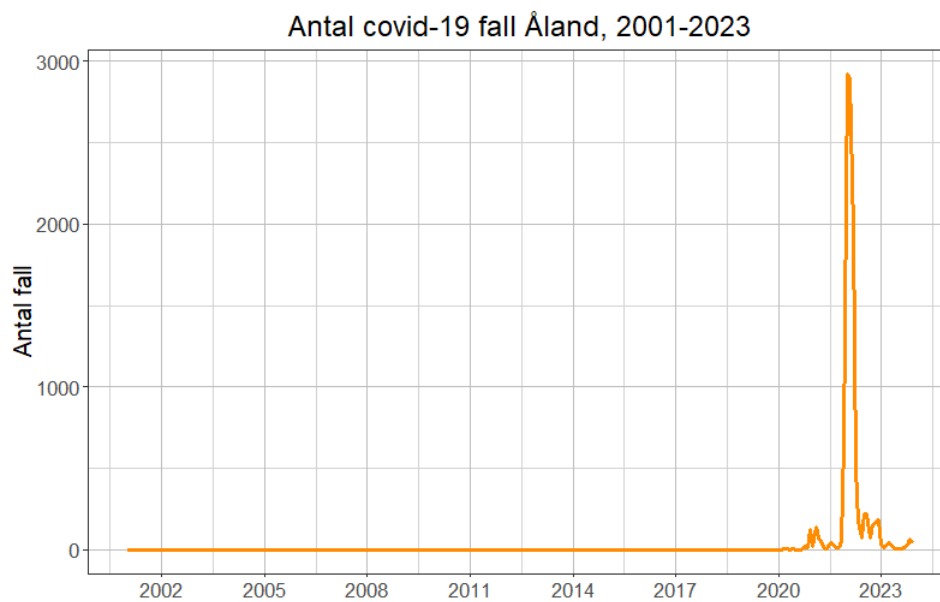
Källa: Ålands statistik- och utredningsbyrå (ÅSUB)

Figur 2.7: Näringslivets totala omsättning, 2001-2023

I figur 2.7 syns det att de totala omsättningarna successivt ökat genom åren, med en viss avvikelse för covid-19 vid 2020, men som sedan hämtat sig och vid 2023 fortsatt verkar öka i nivå, det syns att omsättningarna är som högst i slutet av 2022.

## 2.8 Antal covid-19 fall

Antal covid-19 fall på Åland är en datamängd av tidsseriekaraktär hämtad från THL [10]. Tidsserien beskriver antalet registrerade covid-19 fall på Åland genom åren. Data har bearbetats och beskrivits så att endast aktuell variabel och tidsperiod används.



Källa: Terveysten ja hyvinvoinnin laitos (THL)

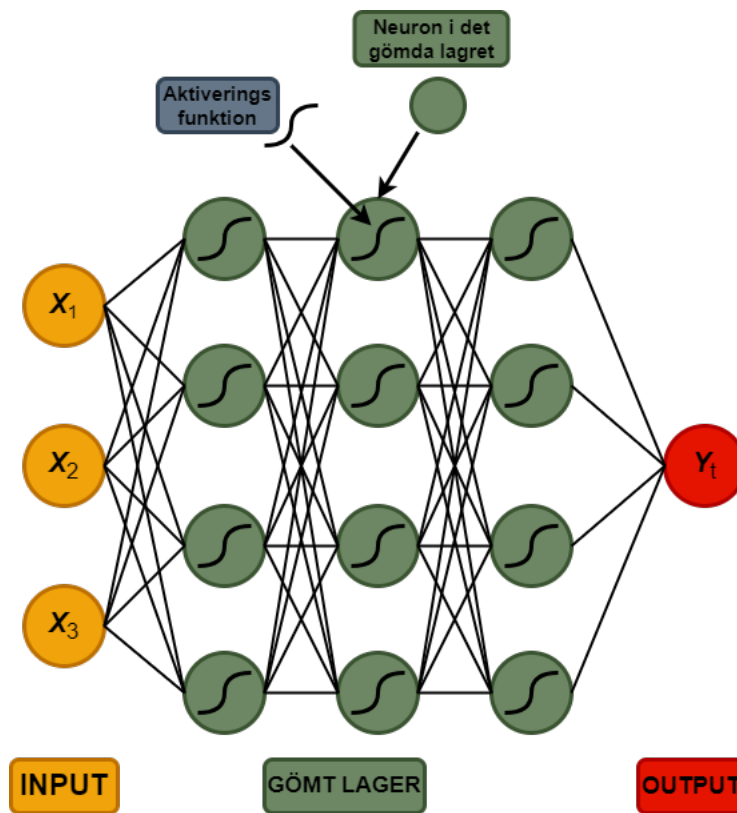
Figur 2.8: Antal covid-19 fall, 2001-2023

I figur 2.8 syns nu att hur antalet fall av covid-19 fluktuerat på Åland genom åren, där det syns att de första registrerade fallen sker efter 2020 och börjar sedan ta fart vid 2021, där det sedan vid 2023 går ner till att vara 0 eller mycket få bekräftade fall av covid-19.



### 3. Metod

Neurala nätverk introducerades först på 1980-talet och har idag utvecklats till en av grundpelarna inom maskininläring. I grunden är ett neuralt nätverk en generalisering av regression vars struktur går ut på att två eller fler lager tillsammans predikterar en responsvariabel  $Y$ . Det som skiljer det neurala nätverket från klassisk regression är att man med en aktiveringsfunktion i det gömda lagret skapar icke-linjäritet i modellen. Detta för att nätverket successivt ska kunna lära sig att lösa komplexa uppgifter, så som regression, mönsterigenkänning, klassificering och även kunna göra prognoser.



Figur 3.1: Neuralt nätverk

I figur 3.1, inspiration [17], synes nu hur ett neuralt nätverk fungerar i praktiken. Det börjar från vänster

där en inmatning av data går igenom det första lagret som här kännetecknas av gula neuroner, från det gula lagret via linjerna mellan neuronerna bestämmer sedan vikter och bias hur mycket inflytande varje datapunkt kommer att ha fram tills att det sedan hamnar i det gömda lagret som kännetecknas av gröna neuroner. I det gömda lagret kommer data sedan att gå igenom en aktiveringsfunktion, här har Sigmoid använts som exempel då det är en vanligt förekommande aktiveringsfunktion. Efter detta går data in i outputlagret där nätverket till sist matar ut en skattning på responsvariabeln.

Vidare kan formeln för neurala nätverk uttryckas enligt:

$$\mathbf{f}_{(x)} = \beta_0 + \sum_{k=1}^K \beta_k h_k(X) \quad (3.1)$$

och med en aktiveringsfunktion som kan beskrivas som en transformation av den input av data som kommer in i nätverkets gömda lager och uttrycks enligt:

$$\mathbf{h}_k = g(w_{k0} + \sum_{j=1}^p w_{kj} X_j) \quad (3.2)$$

där

$\beta_0, \dots, \beta_k$  är skattade parametrar i nätverket.

$X$  är de förklarande variablerna som går in i nätverket.

$h_k$  är en icke-linjär aktiveringsfunktion.

$w_{k0}, \dots, w_{kj}$  är viktmatriser i nätverket.

I Figur 3.1 består det neurala nätverket av tre olika lager men där de gömda lagren är tre stycken i antal. Detta kallas för "Multilayer Perceptron" eller MLP som är en typ av neuralt nätverk där fler gömda lager förekommer. I en MLP används oftast en arkitektur som kallas för feed-forward, som går ut på att informationen i nätverket endast slussas framåt [18]. I det neurala nätverkets arkitektur förekommer det neuroner [19], som inspirerats från den mänskliga hjärnan, neuronernas uppgift i nätverket är att bearbeta och överföra inmatad information. Neuronerna består av tre centrala delar, en eller fler viktade ingångar, en summeringsfunktion för de viktade ingångarna och slutligen en aktiveringsfunktion, som avgör ifall neuronerna ska aktiveras eller inte baserat på summeringens resultat. Om en neuron aktiveras så skickas en signal till de andra anslutna neuronerna i nätverket, på så vis, med den arkitektur som beskrivet ovan kan det neurala nätverket successivt lära sig att lösa komplexa uppgifter. Dessa neuroner förekommer i olika lager, i detta fall tre stycken, där neuronerna har olika uppgifter beroende på i vilket lager de befinner sig. De tre lagren består i ett inmatningslager, ett eller fler gömda lager och sist ett utmatningslager [17]. Inmatningslagret består av en vektor med  $p$  variabler  $X = (X_1, \dots, X_p)$ . I det gömda lagret förekommer  $K$  stycken gömda neuroner med en aktiveringsfunktion. I inmatningslagret matas sedan varje enskilt  $p$  variabler in i det gömda lagrets  $K$  enskilda enheter där olika vikter mellan två neuroner bestämmer hur mycket påverkan inmatningen kommer att ha på den slutliga prediktionen. Därefter går inmatningen genom varje  $K$  enskild neuron från det gömda lagret in i utmatningslagret där ett bias adderas till varje datapunkt för att säkerställa att neuronerna aktiveras även ifall inmatningen är noll. Tillsammans bygger de sedan en icke-linjär funktion  $f(x)$  som skapar prediktionen av  $Y$ .



## 3.1 Aktiveringsfunktioner

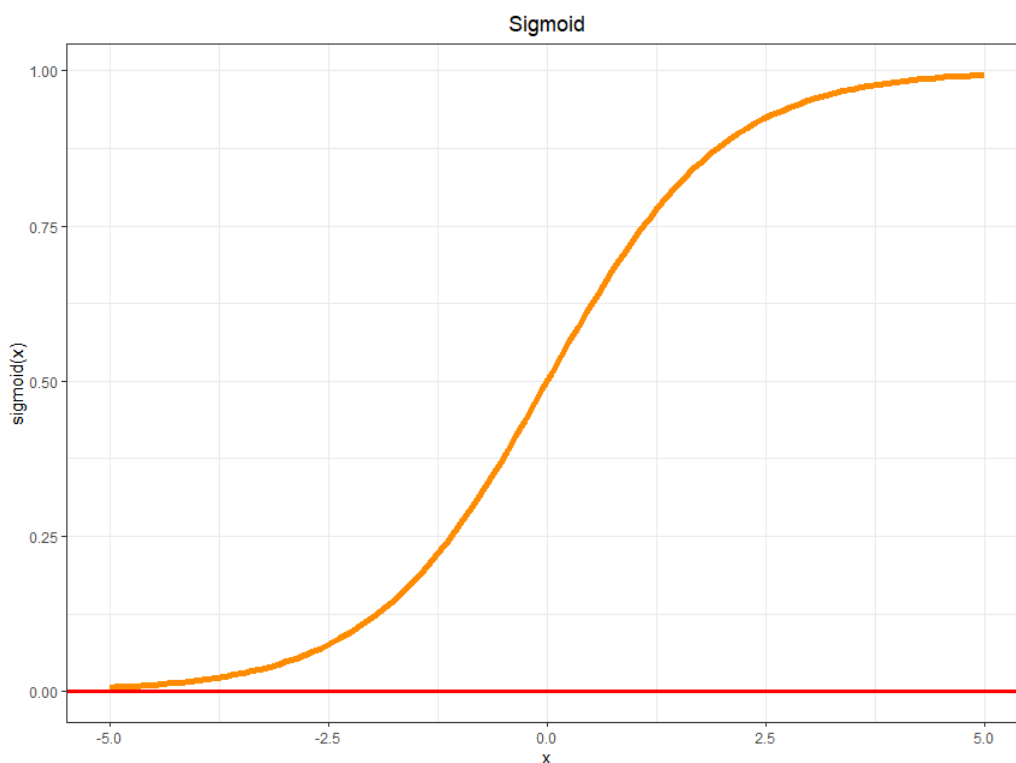
Aktiveringsfunktioner är en central del i det neurala nätverkets arkitektur. Aktiveringsfunktioner används inom neurala nätverk för att transformera inmatning till utmatning genom de olika lagrena i modellen men även för att skapa en icke-linjäritet i modellen. [20] Genom att en aktiveringsfunktion tillämpas så kan modellen upptäcka och lösa mer komplexa problem. Vidare beror det neurala nätverkets träffsäkerhet i prediktionerna på vilken typ av aktiveringsfunktion som används. Det finns flera olika aktiveringsfunktioner.

### 3.1.1 Sigmoid

Sigmoid är en aktiveringsfunktion vars egenskaper bland annat utgörs av icke-linjäritet. En annan av sigmoids egenskaper är att den returnerar värden nära 0 för mindre värden ( $< -5$ ) och nära 1 för större ( $> 5$ ) för att sedan vid utmatning avrunda till exakt 0 eller 1 ( $< -5 = 0$ ) och ( $> 5 = 1$ ). Ett problem som kan identifieras med sigmoidfunktionen är att den inte är centrerad kring origo, vilket gör att alla utgående värden från output får samma tecken, men som genom standardisering av funktionen löser problemet.

Sigmoid definieras enligt följande:

$$\sigma(\mathbf{x}) = \frac{1}{e^{-x}} \quad (3.3)$$



Figur 3.2: Aktiveringsfunktionen Sigmoid

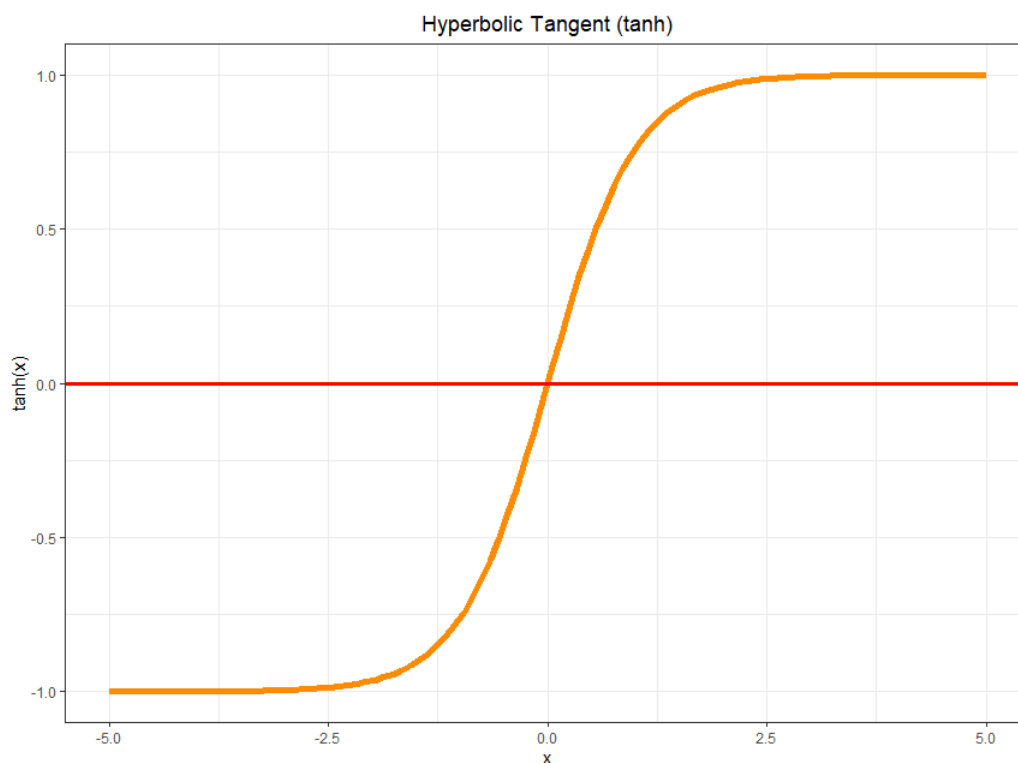
I figur 3.2 synes nu vad som sker ifall en input  $-5 > x > 5$ , om  $x < -5$  kommer Sigmoid att returnera en output som går mot 0 och om  $x > 5$  kommer Sigmoid returnera en output som går mot 1.

### 3.1.2 Tanh

Tanh (hyperbolic tangent) är en aktiveringsfunktion liknande sigmoid. Det som skiljer tanh från sigmoid är att den är centrerad kring origo, vilket genererar andra utmatningar från föregående lager till nästkommande. Tanh är kontinuerlig och differentierbar samt antar värden mellan -1 och 1. Gradienten för tanh är dessutom brantare än för sigmoid och är som aktiveringsfunktion att föredra då även gradienterna, till skillnad från Sigmoid inte är begränsade att variera i en viss riktning men även för att tanh är centrerad kring origo.

Vidare kan tanh definieras enligt följande:

$$\tanh(\mathbf{x}) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \quad (3.4)$$



Figur 3.3: Aktiveringsfunktionen tanh

Likt Sigmoid syns nu här vad som sker i praktiken med tanh som aktiveringsfunktion. Ifall en input  $-2.5 > x > 2.5$  kommer tanh att returnera ett värde som går mot  $-1$  om  $x < -2.5$  och  $1$  om  $x > 2.5$ .

## 3.2 Batch

En batch är en utvald delmängd av träningsdata som används för att uppdatera parametrarna i en modell under en iteration av optimeringsalgoritmen. Istället för att uppdatera parametrarna efter varje enskild datapunkt, används istället batchträning för att minska beräkningskostnaderna och förbättra stabiliteten i träningsprocessen.

Storleken på batchen kan variera beroende på modellens arkitektur, den tillgängliga datorresursen och andra faktorer.

## 3.3 Backpropagation

Backpropagation är en central algoritm inom träning av neurala nätverk. Det är en metod för att beräkna gradienten av förlustfunktionen med avseende på varje parameter i nätverket, vilket gör det möjligt att justera parametrarna för att minimera förlusten under träningen. [17]

Processen börjar med att det matas in ett ingående data i det neurala nätverket, vilket genererar en prediktion. Därefter jämförs prediktionen med de faktiska värdena genom att använda en förlustfunktion, vilket resulterar i en kvantifierad skillnad mellan förutsägelsen och det önskade resultatet. Backpropagation går sedan igenom nätverket bakifrån, från output till input för att beräkna gradienten av förlustfunktionen med avseende på varje parameter. Detta görs genom att tillämpa kedjeregeln för partiella derivator i varje lager av nätverket. Gradienten används sedan i en optimeringsalgoritm, såsom till exempel gradient descent, för att uppdatera parametrarna och minska förlusten under nästa iteration av träningen.

För att kunna uppdatera parametrarna med gradienterna behövs även en inlärningshastighet  $\rho$  anges. Inlärningshastigheten bestämmer hur stora steg gradienterna ska gå mot minimumet av förlustfunktionen.

Om  $\theta$  är en vektor som representerar alla parametrar i det neurala nätverket, så strävar modellen efter att minimera förlustfunktionen  $L$  (i detta fall MSE):

$$L(\theta) = \frac{1}{2} \sum_{i=1}^n (y_i - \hat{y}_i)^2 \quad (3.5)$$

För att nätverket ska veta åt vilket håll det ska justera parametrarna för att minimera förlusten, beräknas gradienten enligt:

$$\nabla L(\theta^m) = \frac{\partial L(\theta)}{\partial \theta} \quad (3.6)$$

Sedan uppdateras parametrarna med hänsyn till gradienten och inlärningshastigheten  $\rho$ :

$$\theta^{m+1} \leftarrow \theta^m - \rho \nabla L(\theta^m) \quad (3.7)$$

Detta är den typiska processen för att träna ett neuralt nätverk med gradientavstigning.

### 3.4 Optimering

Det finns olika metoder och algoritmer för att underlätta optimeringen av neurala nätverk. Nedan kommer de två optimeringsmetoder som använts att presenteras och beskrivas hur de fungerar.

Gradient descent är en grundläggande optimeringsalgoritm som används för att minimera en förlustfunktion och därigenom hitta den optimala lösningen för ett givet problem inom maskininlärning och andra områden. Algoritmen fungerar genom att iterativt justera parametrarna i en modell för att minska värdet på förlustfunktionen.

Processen börjar med att slumpmässigt initialisera parametrarna i modellen. Därefter beräknas gradienten av förlustfunktionen med avseende på varje parameter, vilket ger information om hur mycket och i vilken riktning förlusten förväntas förändras när parametrarna justeras. Gradienten pekar i riktningen av den snabbaste ökningen av förlusten.

Gradient descent använder sedan denna gradientinformation för att uppdatera parametrarna i modellen genom att subtrahera en liten andel av gradienten från varje parameter. Denna andel, känd som inlärningshastigheten eller steglängden, styr hur stora stegen är som tas i parameterrummet vid varje iteration. Genom att upprepa denna process iterativt konvergerar algoritmen mot en lokal minimumpunkt av förlustfunktionen, där förändringen i förlusten är minimal. En utveckling av gradient descent är optimeringsalgoritmen ADAM.

### 3.4.1 ADAM

ADAM (Adaptive Moment Estimation) är en avancerad optimeringsalgoritm som används inom maskininlärning och deep learning för att effektivt uppdatera parametrar i neurala nätverk. Den utvecklades för att kombinera fördelarna med två andra populära optimeringsmetoder: AdaGrad och RMSprop [21]. Dessa kommer kort redovisas nedan. Generella formler för AdaGrad, RMSprop och ADAM kan hittas i [22].

RMSprop, vilket står för Root Mean Square Propagation, är en optimeringsalgoritm som används inom maskininlärning och djupinlärning. Det är en förbättring av den ursprungliga gradientdescenten och syftar till att lösa dess problem med långsam konvergens vid varierande gradienter. RMSprop fungerar genom att justera inlärningshastigheten för varje parameter baserat på ett glidande medelvärde av gradienternas kvadrater. Genom att adaptivt anpassa inlärningshastigheten kan RMSprop effektivt hantera gradienter av olika storlek och därigenom accelerera konvergensen mot en optimal lösning [22].

AdaGrad, vilket står för Adaptive Gradient Algorithm, är en annan optimeringsalgoritm som kan användas. Den utmärker sig genom sin förmåga att justera inlärningshastigheten för varje parameter baserat på historiken av gradienternas magnitud. AdaGrad fungerar genom att tilldela större inlärningshastigheter till sällsynta parametrar och mindre inlärningshastigheter till vanliga parametrar, vilket möjliggör en effektiv konvergens mot en optimal lösning. Denna anpassningsbara inlärningshastighet gör AdaGrad särskilt lämplig för problem med stora datamängder och ojämnt fördelade gradienter. [22]

Den grundläggande idén bakom ADAM är att det använder första och andra momentuppskattningar för att justera inlärningshastigheten för varje parameter. [21] Första momentet är en uppskattning av genomsnittet av gradienterna, medan det andra momentet är en uppskattning av genomsnittet av kvadraten på gradienterna. Dessa uppskattningar används sedan för att uppdatera parametrarna genom att justera inlärningshastigheten för varje parameter baserat på dess gradient och historik. [22]

En av fördelarna med ADAM är dess förmåga att anpassa inlärningshastigheten dynamiskt för varje parameter. [22] Det innebär att parametrar som har sällsynta eller stora gradienter kommer att ha en lägre inlärningshastighet, medan parametrar med vanliga eller små gradienter kommer att ha en högre inlärningshastighet. Detta gör att ADAM kan konvergera snabbt mot en optimal lösning även i problem med olika skalor för gradienter.

## 3.5 Regularisering

En av de utmaningar som neurala nätverk står inför är risken för överanpassning. Under träningsprocessen exponeras modellen för datamaterialet och formas därefter. En optimal modell bör kunna generalisera väl. Problemet uppstår när en komplex modell lär sig datamaterialet extremt noggrant men misslyckas med att generalisera till liknande ny data. Detta fenomen är känt som överanpassning.

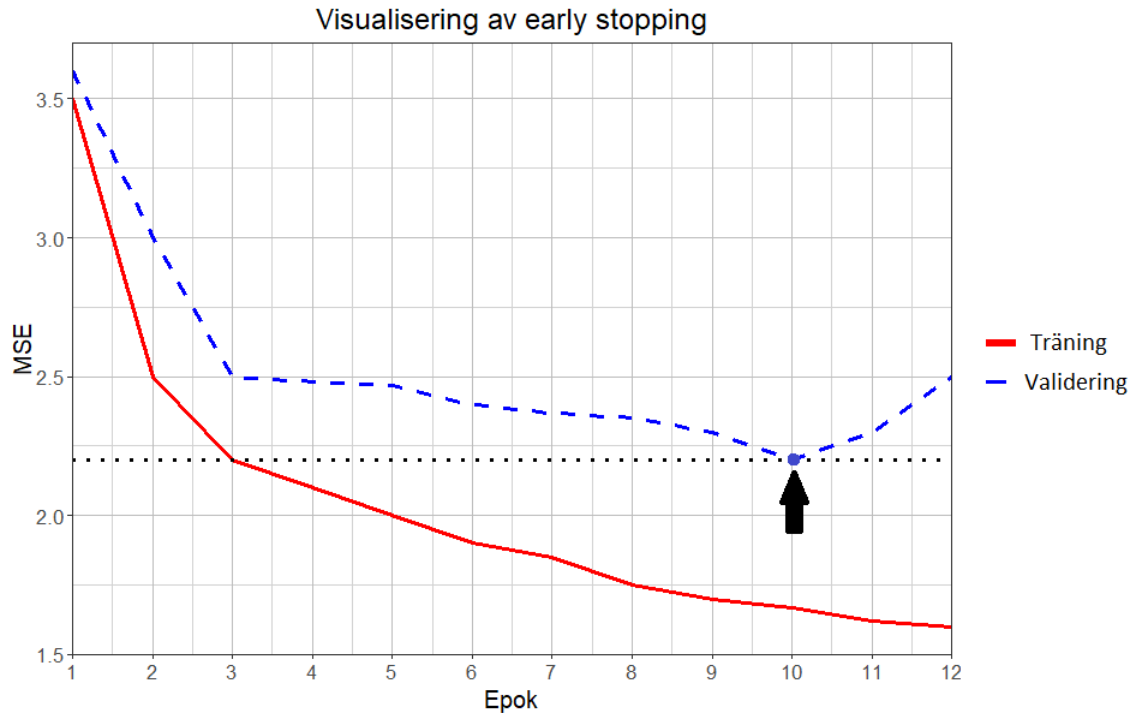
Överanpassning inträffar när modellen anpassar sig för nära det tillgängliga datamaterialet, vilket kan ha en negativ inverkan när modellen tillämpas på nya data. Det innebär att modellen blir känslig för slumpmässigt brus och variationer i det tillgängliga datamaterialet.

Å andra sidan kan modellen ha svårigheter att generalisera överhuvudtaget, vilket indikerar underanpassning. Detta är dock mindre vanligt förekommande.

För att förebygga och övervaka överanpassning kan datamaterialet delas upp i träning-, validering- och testmängder. Vanligtvis används en betydande del av datamaterialet för träning, medan modellen anpassas flera gånger på valideringsdata under träningen för att minimera exempelvis MSE. Sedan kan testdata användas för att utvärdera modellens prestanda på data som den inte tidigare kommit i kontakt med.

### 3.5.1 Early stopping

En effektiv strategi för att förhindra överanpassning är att använda **early stopping** [22]. Genom att övervaka ett utvärderingsmått, såsom MSE, under träningens gång kan modellen optimeras. Detta utvärderingsmått beräknas kontinuerligt, både för tränings- och valideringsdata [22]. Överanpassning kan ofta upptäckas när utvärderingsmättet för träningsdata fortsätter att minska medan måttet för valideringsdata börjar öka. Early stopping är en teknik där en gräns sätts för antalet epoker där utvärderingsmättet inte förbättras. När denna gräns nås avbryts träningen tidigt för att förhindra överanpassning [22]. Genom att stanna träningen vid minsta valideringsfel undviks det att modellen överanpassas till träningsdatan [22].



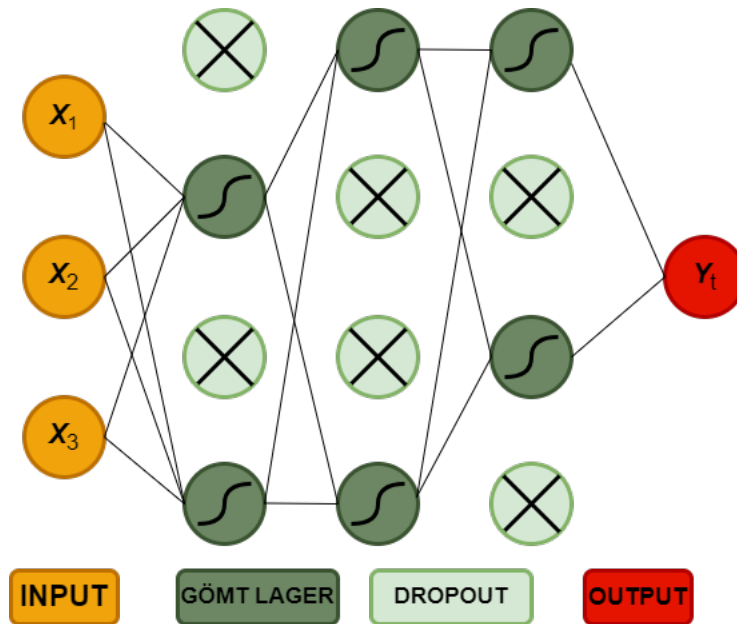
Figur 3.4: Early Stopping visualiserat

I figur 3.4 syns nu hur early stopping fungerar, i diagrammet syns hur data tränas och valideras genom den röda respektive blåa linjen. Vid epok 10 syns sedan hur MSE når det hittills lägsta värdet i körningen och ökar därefter. Då early stopping här är inställd på att stoppas efter n antal epoker, givet att MSE inte fått något lägre värde och sedan gå tillbaka till den epok med lägst MSE, stoppas nu körningen och går sedan tillbaka till epok 10 där det lägsta MSE observerades och därefter kommer data att predikteras från epok 10 och bakåt.

### 3.5.2 Dropout

Dropout är en vanlig regulariseringsmetod inom maskininlärning. Dropout implementeras genom att vid träning, slumpmässigt sätta ett antal  $p$  vikter till 0. Alltså stängs neuronerna av. Detta gör så att nätverket

tvingas att inte lita på några specifika kopplingar, och därmed anpassar nätverket sin kapacitet över hela nätverket istället och därmed motverkar överanpassning. Dropout sker endast vid träning av nätverket och efter träning stängs dropout av. Dropout kan anpassas på ett eller flera gömda lager åt gången.



Figur 3.5: Dropout visualiserat i ett neuralt nätverk

I figur 3.5 syns ett neuralt nätverk där dropout har anpassats en gång. Dropoutlager har anpassats för varje gömt lager med en dropoutandel på 50 % ( $p=0.5$ ). De neuroner med kryss i sig har blivit tillgivna vikter som är 0 och därmed blir dessa neuroner avstängda. Olika avstängningar av dropout kommer att anpassas under träningen, alltså att olika vikter till neuronerna sätts till 0.

En vidare utveckling av detta presenteras senare i metodkapitlet med MC Dropout.

## 3.6 Osäkerheter

För att kunna styrka om en modell är bra eller dålig kan osäkerheten användas. Normalt tas denna inte hänsyn till i neurala nätverk, utan de behöver specifikt modelleras in och detta för att ta hänsyn till osäkerheten. Det finns två typer av osäkerheter, **epistemisk** och **aleatorisk**. Den epistemiska osäkerheten är den osäkerhet som kommer från modellen medan den aleatoriska är den osäkerhet som kommer från observationerna. [23]

### 3.6.1 Epistemisk osäkerhet

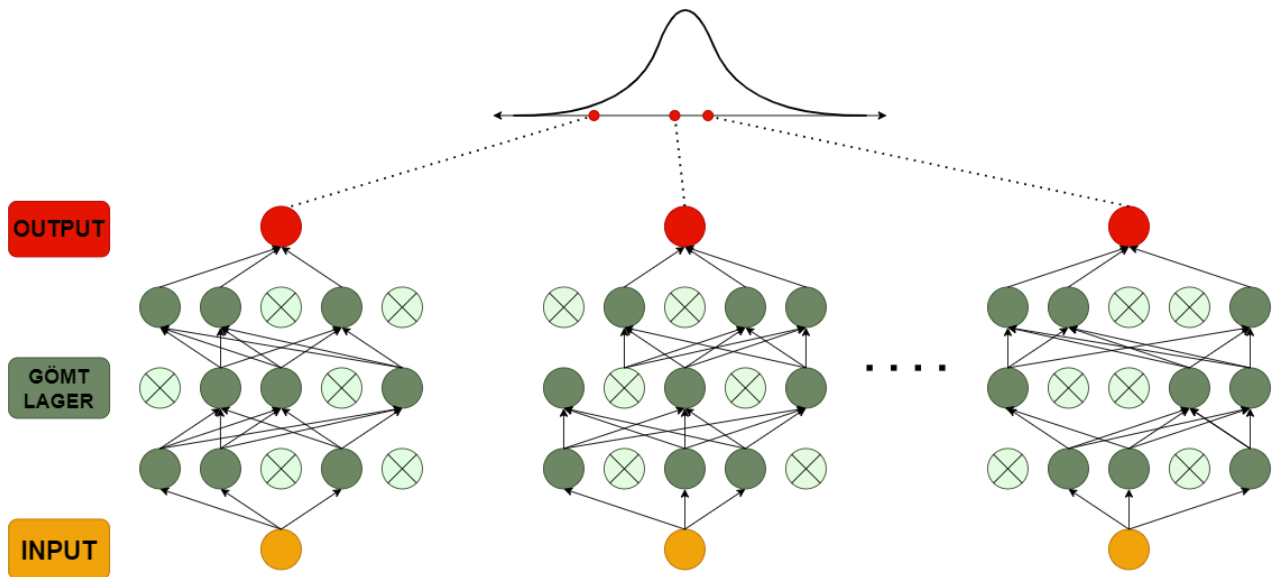
Som ovan nämnt är den epistemiska osäkerheten den osäkerhet som kommer från modellen. För att få ut detta används normalt sett Bayesianska neurala nätverk (BNN). Målet för BNN är att istället för att hitta en punktskattning för parametrar i modellen, t.ex. vikter, skapa fördelningar för varje parameter. Därmed

kommer alla parametrar ha en osäkerhet som kommer påverka vad som kommer matas ut ur nätverket. Antalet parametrar i ett BNN är huvudfaktorn som gör att det inte alltid är rimligt att använda. Eftersom en fördelning skapas för varje parameter skattas i praktiken varje parameter många gånger om vilket kräver en större uträkningskraft, vilket privata användare oftast inte har tillgång till.

Gal et al. [24] har tagit fram en metod som kallas **Monte-Carlo Dropout** (MC-Dropout), som gör detta utan att behöva skatta ett BNN. Detta sker genom att skapa ett dropout nätverk. Alltså ett neuralt nätverk där ett dropoutlager mellan varje viktlager används. De visar då genom användning av en djup gaussisk process att då ett dropoutnätverk skattas är detta matematiskt ekvivalent med Bayesiansk inferens [22].

När modellen körs på träningsdata så fungerar dropout som normalt, därefter, när träningsdata körts igenom och modellen sedan körs på nya data så kommer dropoutlagrena från träningen att lämnas på så att varje neuron i dropoutlagret slumpmässigt stängs av genom alla körningar. Tack vare detta kommer nätverket använda olika neuronbanor varje gång den körs. För att sedan få ett mått på osäkerheten tillkallas modellen flera gånger medan resultaten sparas för varje körning. Sedan beräknas medelvärdet för alla körningar för varje motsvarande punkt i prediktionerna, där  $M$  är totala antalet körningar av modellen,  $f(x, p)$  är en framåtsvepning av nätverket, och  $p$  är andelen dropout i varje dropout lager. [25]

$$\hat{y} = \frac{1}{M} \sum_{m=1}^M f(x, p) \quad (3.8)$$



Figur 3.6: MC-Dropout visualiserat

I figur 3.6 syns nu hur ett nätverk med MC-Dropout fungerar i praktiken. Samma neurala nätverk tillkallas  $M$  gånger för med olika masker av dropout på alla gömda lager. De avstängda neuronerna, alltså de neuronerna som har vikter med 0, är de neuronerna med ett kryss i och har en ljusare nyans av grönt. Efter alla körningar skapas en fördelning för prediktionerna. Detta syns också i figur 3.6.



### 3.6.2 Aleatorisk Osäkerhet

Som tidigare nämnt ovan är den aleatoriska osäkerheten den osäkerhet som kommer från observationerna som modellen tränas på. [25]. Detta brukar normalt sett delas in i två delgrupper: homo-, och heteroskedastisk osäkerhet. För homoskedastisk osäkerhet antas variansen vara konstant för alla observationer, medan vid heteroskedastisk osäkerhet antas variansen skilja sig åt beroende på input. [23] I detta arbete kommer heteroskedastisk osäkerhet antas då variansen ej antas vara konstant. Metoden för heteroskedastisk osäkerhet går ut på att träna det neurala nätverket för att minimera den Gaussiska negativa log likelihooden som en förlustfunktion. För att kunna göra detta måste nätverkets output kunna antas följa en normalfördelning.

För att göra detta rekommenderas det enligt Kendall och Gal [23] att träna nätverket genom att även prediktera logvarianserna  $s_t = \log \hat{\sigma}_t^2$ , så att outputen från nätverket blir enligt:

$$(\hat{y}_1, \hat{y}_2, \dots, \hat{y}_{24}, \log \hat{\sigma}_1^2, \log \hat{\sigma}_2^2, \dots, \log \hat{\sigma}_{24}^2)$$

Logvarianserna skattas för att undvika prediktioner för variansen som är noll eller negativa. [25]

För att nu ta i beaktande för aleatoriska osäkerheten så vill en minimering av den Gaussiska negativa log-likelihood (GNLL) förlustfunktionen göras [25] enligt:

$$L_{nn}(\theta) = \frac{1}{2D} \sum_i \exp(-s_i) \|y_i - \hat{y}_i\|^2 + s_i \quad (3.9)$$

### 3.6.3 Kombination epistemisk och aleatorisk osäkerhet

Enligt Kendall (2017) [23] går det även att kombinera dessa två typer av osäkerheter. Detta görs genom att använda någon metod för att ta i beaktande den epistemiska osäkerheten, förslagsvis MC-dropout, medan den aleatoriska osäkerheten även tas i beaktande som beskrivet ovan. När detta görs går det även att skatta den prediktiva osäkerheten genom [25]:

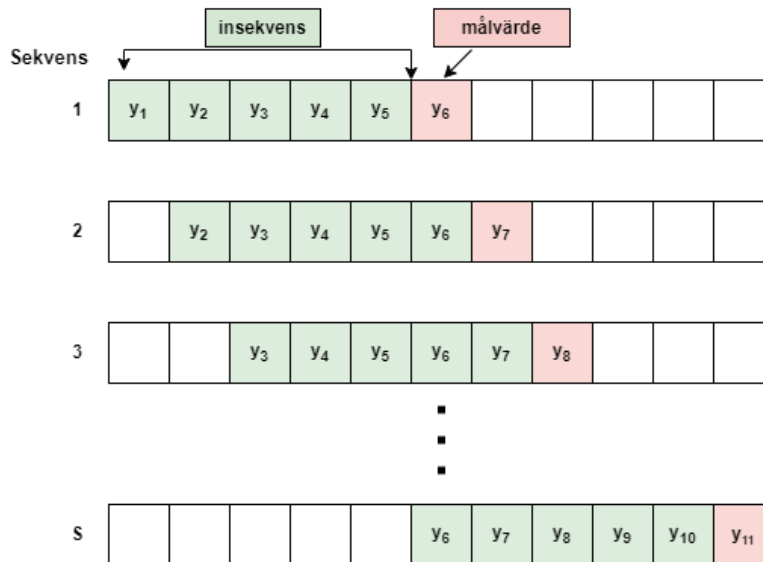
$$Var(y) \approx \frac{1}{T} \sum_{t=1}^T \hat{y}_t^2 - \left(\frac{1}{T} \sum_{t=1}^T \hat{y}_t\right)^2 + \frac{1}{T} \sum_{t=1}^T \hat{\sigma}_t^2 \quad (3.10)$$

## 3.7 Recurrent Neural Network

Ett Recurrent Neural Network (RNN), eller på svenska rekursivt neuralt nätverk, är en anpassning av ett neuralt nätverk som klarar av att hantera olika typer av sekvenser av data, såsom tidsseriedata, text, ljud, och använder detta för olika ändamål. Ett exempel är självkörande bilar, där en kombination av ett faltat- och rekursivt neuralt nätverk kan försöka förutsäga bilens körväg. [25] [22] [17]. Det finns olika typer av RNNs; en-till-en (one-to-one), en-till-flera (one-to-many), flera-till-en (many-to-one) och flera-till-flera (many-to-many). En-till-en är då en input används för att få en output. En-till-flera betyder att en input används för att få ut en sekvens av outputs. I flera-till-en används flera inputs för att få en output. I flera-till-flera används flera inputvärden eller en sekvens av dem för att få ut flera eller en sekvens för output.

### 3.7.1 Input för RNN

För att RNN skall kunna användas behöver data bearbetas till en viss struktur. Den metod för att göra detta kallas för fönstring (windowing). Först väljs en fönsterstorlek, t.ex. 5. De första fem observationerna kommer då användas som första inputfönster. Sedan glider fönstret en tidpunkt längre fram och tar nästkommande fem värden. Detta görs genom hela datamaterialet. Medan detta sker görs även en fönstring för målsekvensen som är ett tidssteg ytterligare förskjutet jämfört med insekvensen, detta görs även genom hela datamaterialet. Så första fem tidsstegen är en input och det sjätte tidssteget är det som ska predikteras (output/målvärde). Det går även att använda flera variabler för att prediktera målvärdet. Då har varje variabel samma fönsterstorlek på insekvensen för att prediktera målvärdet (output). Alltså används då flera variabler för att prediktera målvärdet.



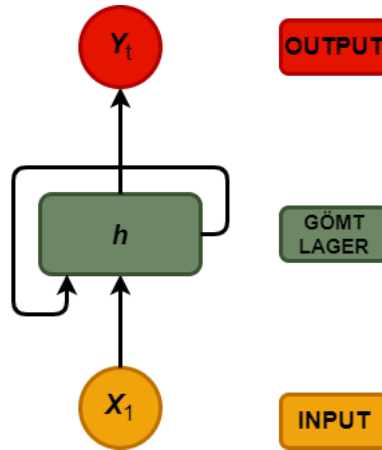
Figur 3.7: Fönstring av en variabel

I figur 3.7 syns nu ett enkelt exempel där fönstring skett på en variabel  $y$ .  $S$  är antalet sekvenser som används för att komma genom hela variabeln. Fönsterstorleken för insekvensen är 5 och längden på målsekvensen är 1.

I vissa fall är inte målet att prediktera endast nästkommande värden. Exempelvis vid användning av sekvens 1, är målet möjligen att prediktera tidssteg 8 istället för tidssteg 6. Detta är fullt möjligt att göra givet att fönstringen av datamaterialet skapas med detta i beaktande. Det går även att prediktera alla värdena mellan tidssteg 5 och tidssteg 8. Då fås en vektor med värden. Likt i fallet för ett värde måste fönstringen då sättas upp på detta vis.

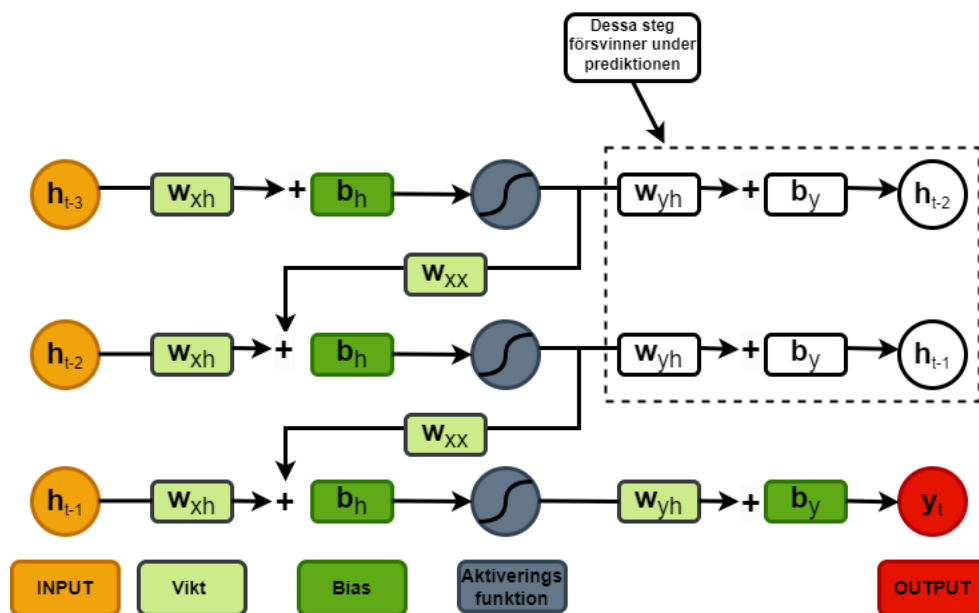
### 3.7.2 Struktur RNN

Generellt kan ett RNN se ut som följande:



Figur 3.8: RNN-struktur med ett gömt lager

Det som nu syns i figur 3.8, inspiration [22], är hur ett RNN fungerar i praktiken, här visas hur nätverket skickar in data  $X_1$  från tidigare tidpunkter in i nätverket som passerar ett gömt lager  $h$  och sedan skickas tillbaka in i  $h$  när nästa tidpunkt i data kommer in i nätverket och fortsätter så genom hela nätverket fram tills att den sista tidpunkten går in i data, där de tillsammans, baserad på deras värden påverkar den output  $Y_t$  som då blir den slutliga prediktionen.



Figur 3.9: Utvecklad RNN

Nu synes i figur 3.9, inspiration [22], en mer detaljerad struktur för RNN, här visas nu hur en input  $h_{t-3}$  gått in i det gömda lagret där den multipliceras med en viktmatris  $w_{xh}$  och sedan adderas med en biasterm  $b_h$  och därefter går igenom en Sigmoid-funktion. Som det syns i den vita streckade rutan så står det ju att de stegen inte används för prediktionen, vilket beror på att flera tidpunkter används och körs in i nätverket, hade det däremot varit så att ändamålet var att skatta  $h_{t-2}$  hade nätverket fortsatt i den riktningen men då målet nu är att använda flera tidpunkter från en serie så stryks dessa steg under prediktionen fram tills att sista tidpunkten matas in. Vad som då sker är att efter att  $h_{-3}$  gått igenom sigmoid så åker den in och adderas med  $h_{t-2}$  som gått in i nätverket och multipliceras med en viktmatris  $w_{xx}$ , adderas med ett bias  $b_h$  och passerar ytterligare en sigmoid innan samma steg repeteras återigen, fram tills att sista tidpunkten i data har gått igenom nätverket, där kommer de likt tidigare steg multipliceras med en viktmatris  $w_{xh}$ , adderas med en biasterm  $b_h$ , gå igenom en sigmoid och nu multipliceras med en ny viktmatris  $w_{yh}$  och adderas med en ny biasterm  $b_{yh}$  och når till sist output  $y_t$  som då blir den slutliga prediktionen.

Vidare kan ett RNN sedan uttryckas enligt följande:

$$\begin{aligned} \mathbf{h}_{(t)} &= g_1(\mathbf{W}_{xh}\mathbf{h}_{t-1} + \mathbf{W}_{xx}\mathbf{x}_t + \mathbf{b}_h) \\ y_{(t)} &= g_2(\mathbf{W}_{yh}\mathbf{h}_t + \mathbf{b}_y) \end{aligned} \quad (3.11)$$

där

$g_1$  och  $g_2$  är aktiveringsfunktioner.

$W$  är en viktmatris.

$b$  är en biasterm.

låt  $\theta = \{W_{yh}, W_{xh}, W_{xx}, b_h, b_y\}$  vara parametrar som kan optimeras.

### 3.7.2.1 Backpropagation Through Time

Likt vanlig bakåtpropagering vill gradienterna hittas. Då flera tidssteg nu är med bör gradienterna beräknas och summeras för varje tidssteg.

Låt  $L$  vara en godtycklig förlustfunktion, exempelvis MSE:

$$L(\theta) = \frac{1}{t} \sum_{t=1}^T (y_t - \hat{y}_t)^2 \quad (3.12)$$

$$\frac{\partial L}{\partial \mathbf{W}_{yh}} = \frac{\partial L}{\partial \hat{y}} \frac{\partial \hat{y}}{\partial \mathbf{W}_{yh}} \quad (3.13)$$

$$\frac{\partial L}{\partial b_y} = \frac{\partial L}{\partial \hat{y}} \frac{\partial \hat{y}}{\partial b_y} \quad (3.14)$$

$$\frac{\partial L_T}{\partial \mathbf{W}_{xh}} = \sum_{t=1}^T \frac{\partial L_T}{\partial \hat{y}_T} \frac{\partial \hat{y}_T}{\partial h_t} \frac{\partial h_t}{\partial \mathbf{W}_{xh}} \quad (3.15)$$

$$\frac{\partial L_T}{\partial \mathbf{W}_{hh}} = \sum_{t=1}^T \frac{\partial L_T}{\partial \hat{y}_t} \frac{\partial \hat{y}_t}{\partial h_t} \frac{\partial h_t}{\partial \mathbf{W}_{hh}} \quad (3.16)$$

$$\frac{\partial L_T}{\partial \mathbf{b}_h} = \sum_{t=1}^T \frac{\partial L_T}{\partial \hat{y}_t} \frac{\partial \hat{y}_t}{\partial h_t} \frac{\partial h_t}{\partial \mathbf{b}_h} \quad (3.17)$$

Sedan används dessa gradienter i samspel med inlärningstakten för att uppdatera parametrarna.

### 3.7.2.2 Problematik med gradient vanishing och gradient explosion

Inom RNN kan det uppstå ett problem med det som kallas för gradient vanishing och gradient exploding som kort kan förklaras som att gradienten antingen försvinner eller exploderar. Det som sker är att i figur 3.6, när en input från en tidigare tidpunkt går igenom en sigmoidfunktion och sedan multipliceras med en vikt  $W_{xx}$  uppstår det problem när  $W_{xx}$  är skild från 1 detta för att  $W_{xx}$  successivt kommer att höjas upp med inputstorleken, låt säga att om inputstorleken är 150 observationer så kommer  $W_{xx}$  att bli  $W_{xx}^{150}$ . Om  $W_{xx} < 1$  så kommer då gradienten successivt att försvinna, om  $W_{xx} > 1$  så kommer gradienten att explodera. Det som då sker är att förlustfunktionen inte kommer att hitta minimum för att den kommer att ta för små steg när den konvergerar när  $W_{xx} < 1$ , eller ta för stora steg när  $W_{xx} > 1$ . Därför används sällan ett RNN som beskrivet ovan då syftet med det arbete som ämnas att utföras oftast innefattar en större inputstorlek och det är där som LSTM- och GRU-celler kommer in då deras struktur behandlar problem med gradient explosion eller gradient vanishing.[26]

### 3.7.2.3 Exempel: Backpropagation Through Time

Backpropagation through time är egentligen väldigt likt vanlig backpropagation. Eftersom ett RNN nu har flera tidpunkter måste detta tas i beaktande vid beräkning av gradienterna. Nedan visas detta i ett förtydligt exempel.

Om ett RNN antas utan biastermer (för enkelhetens skull) skulle ett RNN se ut som följande:

$$\begin{aligned} \mathbf{h}_{(t)} &= g_1(\mathbf{W}_{xh}\mathbf{h}_{t-1} + \mathbf{W}_{xx}\mathbf{x}_t) \\ y_{(t)} &= g_2(\mathbf{W}_{yh}\mathbf{h}_t) \end{aligned} \quad (3.18)$$

där  $g_1$  och  $g_2$  är valfria aktiveringsfunktioner.

Om sedan totalt 3 tidpunkter antas, det vill säga  $t = 3$ . Antas sedan en enkel förlustfunktion som:

$$L(\theta) = (y_t - \hat{y}_t)^2 \quad (3.19)$$

Om gradienten för  $\mathbf{W}_{yh}$  vill beräknas görs det genom:

$$\frac{\partial L_3}{\partial W_{yh}} = \frac{\partial L_3}{\partial \hat{y}_3} \frac{\partial \hat{y}_3}{\partial W_{yh}} \quad (3.20)$$

Om gradienten för  $\mathbf{W}_{xh}$  vill beräknas görs det genom:

$$\frac{\partial L_3}{\partial W_{xh}} = \left( \frac{\partial L_3}{\partial \hat{y}_3} \frac{\partial \hat{y}_3}{\partial h_3} \frac{\partial h_3}{\partial W_{xh}} \right) + \left( \frac{\partial L_3}{\partial \hat{y}_3} \frac{\partial \hat{y}_3}{\partial h_3} \frac{\partial h_3}{\partial h_2} \frac{\partial h_2}{\partial W_{xh}} \right) + \left( \frac{\partial L_3}{\partial \hat{y}_3} \frac{\partial \hat{y}_3}{\partial h_3} \frac{\partial h_3}{\partial h_2} \frac{\partial h_2}{\partial h_1} \frac{\partial h_1}{\partial W_{xh}} \right) \quad (3.21)$$

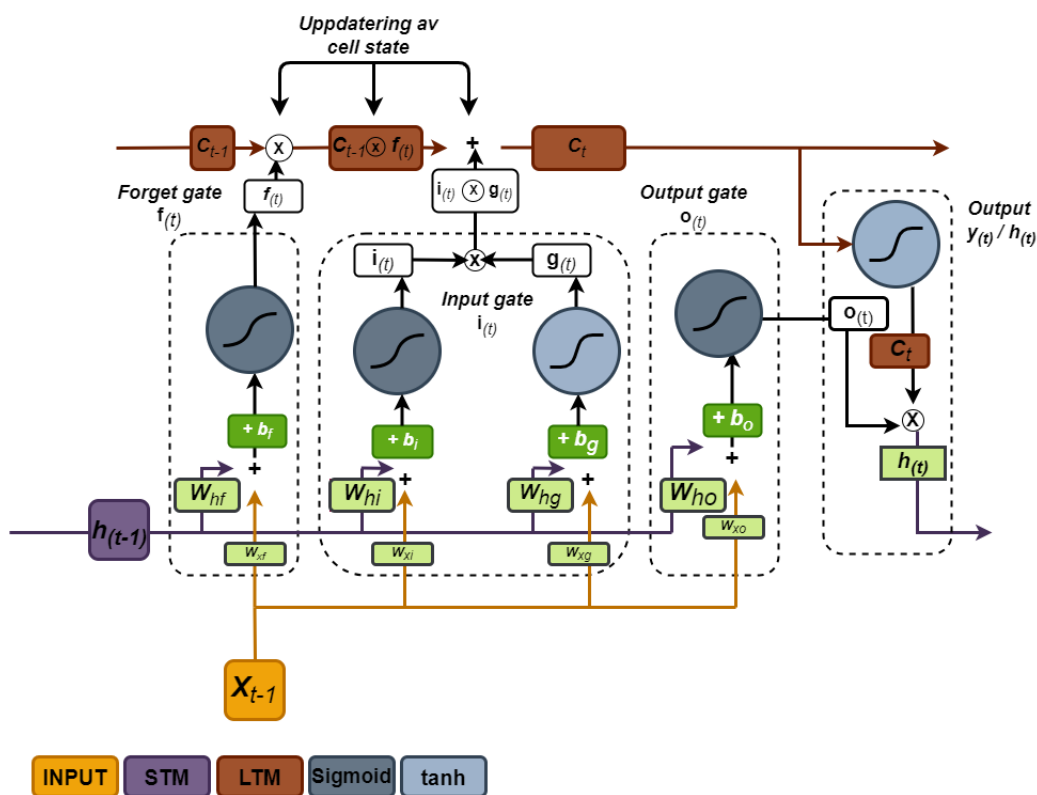
Om gradienten för  $\mathbf{W}_{xx}$  vill beräknas görs det genom:

$$\frac{\partial L_3}{\partial W_{xx}} = \left( \frac{\partial L_3}{\partial \hat{y}_3} \frac{\partial \hat{y}_3}{\partial h_3} \frac{\partial h_3}{\partial W_{xx}} \right) + \left( \frac{\partial L_3}{\partial \hat{y}_3} \frac{\partial \hat{y}_3}{\partial h_3} \frac{\partial h_3}{\partial h_2} \frac{\partial h_2}{\partial W_{xx}} \right) + \left( \frac{\partial L_3}{\partial \hat{y}_3} \frac{\partial \hat{y}_3}{\partial h_3} \frac{\partial h_3}{\partial h_2} \frac{\partial h_2}{\partial h_1} \frac{\partial h_1}{\partial W_{xx}} \right) \quad (3.22)$$

Om biastermerna läggs till går det att utnyttja kedjeregeln på liknande sätt för att få ut vikterna. Gradienterna används sedan, likt ett normal neuralt nätverk, tillsammans med learning rate för att uppdatera parametrarna.

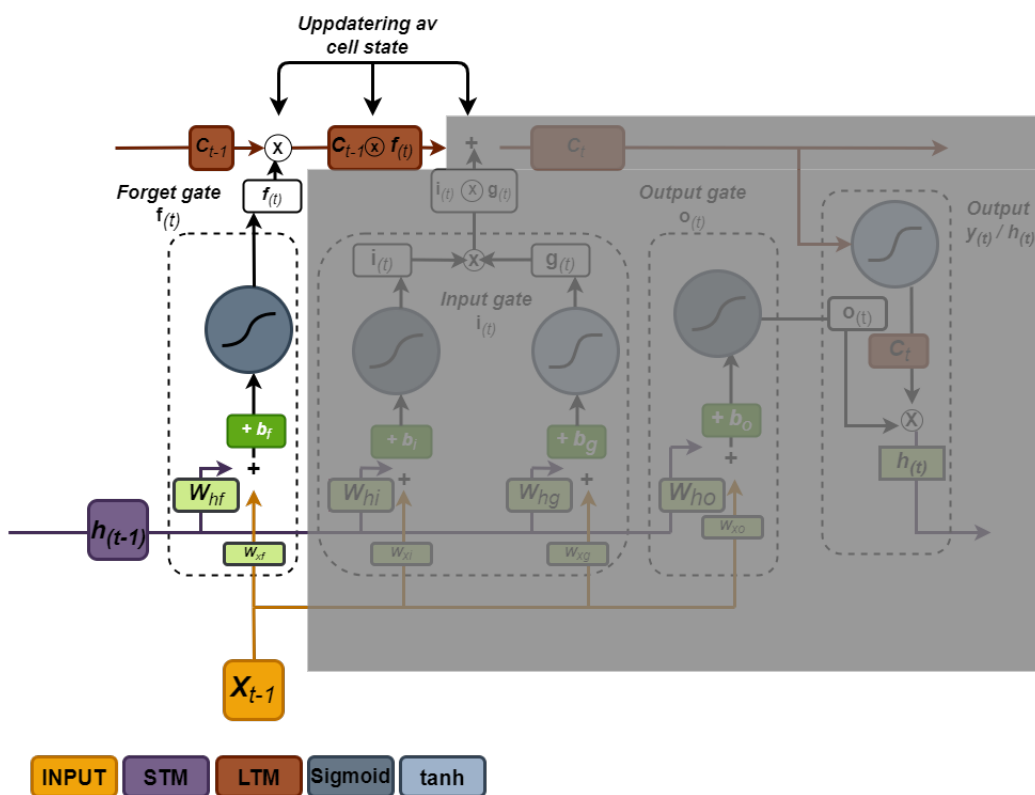
## 3.8 Long Short Term Memory

Long Short Term Memory (LSTM) introducerades av Hochreiter & Schmidhuber år 1997 [26] och har blivit populärt för sin förmåga att hantera långa sekvenser och hantera minnen över längre tidsperioder. Det består av en komplex cellstruktur med tre grindar: forget gate, input gate och output gate. Dessa grindar styr flödet av information genom cellen och möjliggör att LSTM minns relevanta delar av sekvensen samtidigt som den kan glömma irrelevanta delar. Vidare kan det beskrivas som att cellen tar emot tre stycken inflöden av information, dessa är en input som skickar in det data som ska predikteras in i modellen, den andra en cell state, som hanterar cellens långtidsminne och slutligen hidden state som här hanterar korttidsminnet i cellen.



Figur 3.10: Cellstruktur LSTM

I figur 3.10, inspirerad av Géron, [22], visas hur en LSTM-cell fungerar i praktiken. I cellen går det in en input  $X_{(t-1)}$  som är det data som modellen gör sina prediktioner på. Vidare går det in ett korttidsminne  $h_{(t-1)}$  och ett långtidsminne  $C_{(t-1)}$  som båda är 0 vid den första tidpunkten och därefter ett beräknat värde av föregående tidpunkt. Varje gate kännetecknas av att input och korttidsminnet multipliceras med en varsin viktmatris och därefter adderas tillsammans med ett bias och slutligen går igenom en aktiveringsfunktion, som här antingen är sigmoid eller tanh. När input och korttidsminnet gått igenom Forget gate multipliceras det värdet elementvis med värdet från långtidsminnet som sedan adderas med en elementvis produkt av värdena i Input gate och Cell state och då uppdaterar det gamla långtidsminnet som antingen kommer användas för nästa tidpunkt eller för den slutliga prediktionen. Nedan kommer nu varje enskild gate att förklaras i detalj.



Figur 3.11: Forget gate LSTM

$$\mathbf{f}_{(t)} = \sigma(\mathbf{W}_{xf}^\top \mathbf{x}_{(t)} + \mathbf{W}_{hf}^\top \mathbf{h}_{(t-1)} + \mathbf{b}_f) \quad (3.23)$$

där

$\mathbf{f}_{(t)}$  är forget gate.

$\mathbf{W}_{xf}$  är en specifik viktmatris för input i forget gate.

$\mathbf{x}_{(t)}$  är en input av tidpunkt  $t$  i data.

$\mathbf{W}_{hf}$  är en specifik viktmatris från korttidsminnet i forget gate.

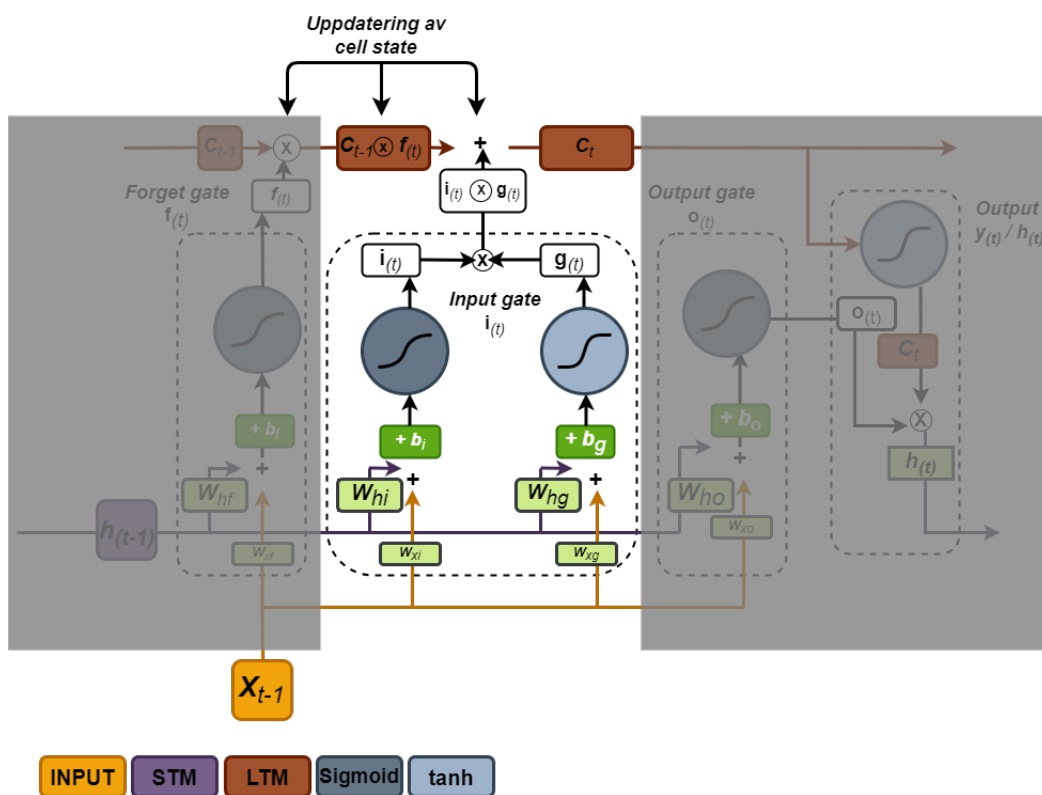
$\mathbf{h}_{(t-1)}$  är hidden state som hanterar korttidsminnet från föregående cell.

$\mathbf{b}_f$  är en specifik biasterm i forget gate.

$\sigma$  aktiveringsfunktionen Sigmoid.

Forget gates huvudsakliga funktion är att beräkna andelen av långtidsminnet som ska tas i beaktande då cell state ska uppdateras. Detta sker genom att input  $X_{t-1}$  och korttidsminnet från hidden state  $h_{(t-1)}$  multipliceras med respektive viktmatriser ( $h_{(t-1)} \cdot W_{hf}$  och  $X_{(t-1)} \cdot W_{xf}$ ) som sedan adderas med en biasterm ( $b_f$ ). Denna beräkning körs sedan in i en aktiveringsfunktion, vanligast en Sigmoid-funktion. Vilket ger  $f_{(t)}$  som multipliceras elementvis med värdena från långtidsminnet i cell state  $C_{(t-1)}$ .





Figur 3.12: Input gate LSTM

$$\begin{aligned}
 \mathbf{i}_{(t)} &= \sigma (\mathbf{W}_{xi}^\top \mathbf{x}_{(t)} + \mathbf{W}_{hi}^\top \mathbf{h}_{(t-1)} + \mathbf{b}_i) \\
 \mathbf{g}_{(t)} &= \tanh (\mathbf{W}_{xg}^\top \mathbf{x}_{(t)} + \mathbf{W}_{hg}^\top \mathbf{h}_{(t-1)} + \mathbf{b}_g) \\
 \mathbf{c}_{(t)} &= \mathbf{f}_{(t)} \otimes \mathbf{c}_{(t-1)} + \mathbf{i}_{(t)} \otimes \mathbf{g}_{(t)}
 \end{aligned}
 \tag{3.24}$$

där

$\mathbf{i}_{(t)}$  är första delen av input gate.

$\mathbf{g}_{(t)}$  är andra delen av input gate.

$\mathbf{W}_{xi}$ ,  $\mathbf{W}_{xg}$  är en specifik viktmatris för första och andra delen av input i input gate.

$\mathbf{x}_{(t)}$  är en input av tidpunkt  $t$  i data.

$\mathbf{W}_{hi}$ ,  $\mathbf{W}_{hg}$  är en specifik viktmatris från korttidsminnet för första och andra delen i input gate.

$\mathbf{b}_i$ ,  $\mathbf{b}_g$  är en specifik biasterm för första och andra delen i input gate.

$\sigma$  är aktiveringsfunktionen Sigmoid.

$\tanh$  är aktiveringsfunktionen tanh

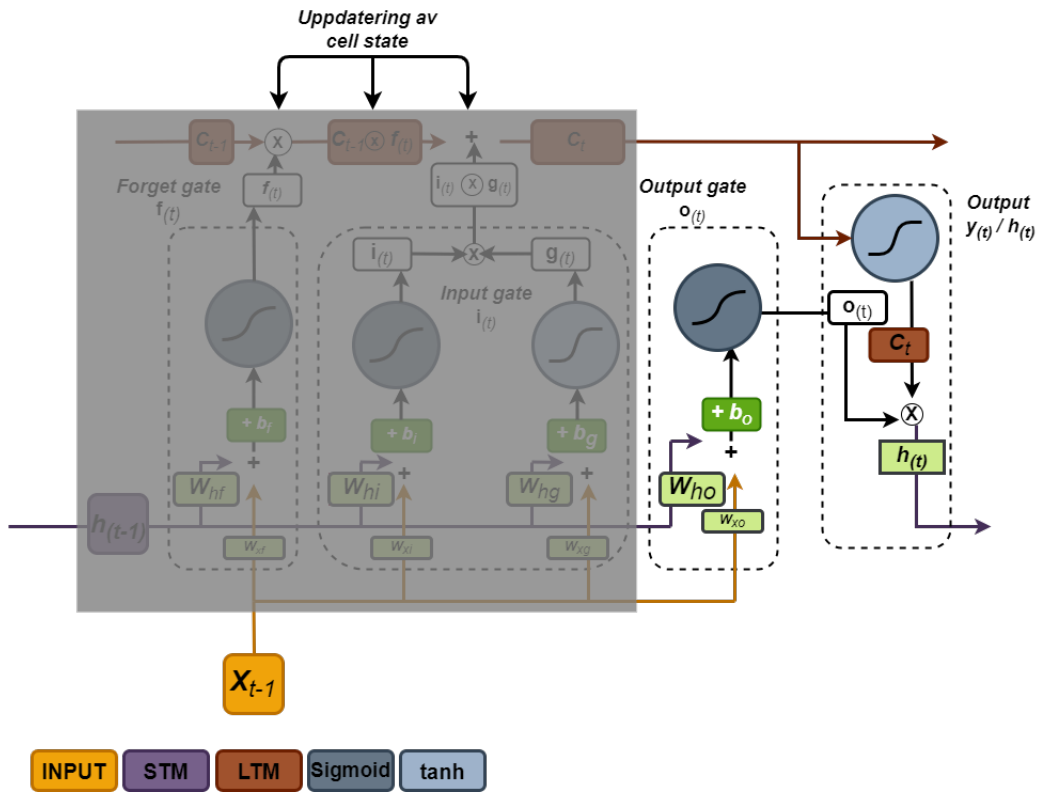
$\mathbf{c}_{(t)}$  är uppdaterade cell state.

$\mathbf{h}_{(t-1)}$  är hidden state som hanterar korttidsminnet från föregående cell.

Input gate består av två delar,  $i_t$  och  $g_t$ , i första delen av input gate  $i_t$  multipliceras korttidsminnet  $h_{(t-1)}$  med en ny viktmatris  $W_{hi}$  och input  $x_t$  med  $W_{xi}$ , därefter adderas de två tillsammans med ett bias  $b_i$  och går sedan igenom en sigmoid-funktion.

Samtidigt i den andra delen av input gate  $g_t$  multipliceras input  $x_t$  med en viktmatris  $W_{xg}$  och korttidsminnet  $h_{(t-1)}$  med  $W_{hg}$  och som sedan adderas ihop tillsammans med ett bias  $b_g$ , där efter går summan genom en tanh-funktion.

Till sist multipliceras input gates två delar  $i_t$  och  $g_t$  med varandra elementvis, där efter adderas de med den elementvisa produkten av forget gate  $f_t$  och långtidsminnet  $c_{t-1}$  och uppdaterar då cell state  $c_t$  som sedan skapar ett nytt långtidsminne som kommer användas i nästkommande LSTM-cell och för en ny tidpunkt i data.



Figur 3.13: Output gate LSTM

$$\begin{aligned} \mathbf{o}_{(t)} &= \sigma(\mathbf{W}_{xo} \top \mathbf{x}_{(t)} + \mathbf{W}_{ho} \top \mathbf{h}_{(t-1)} + \mathbf{b}_o) \\ \mathbf{y}_{(t)} &= \mathbf{h}_{(t)} = \mathbf{o}_{(t)} \otimes \tanh(\mathbf{c}_{(t)}) \end{aligned} \quad (3.25)$$

där

$\mathbf{o}_{(t)}$  är output gate.

$\mathbf{W}_{x_o}$  är en specifik viktmatris för input i output gate.

$\mathbf{x}_{(t)}$  är en input av tidpunkt  $t$  i data.

$\mathbf{W}_{h_o}$  är en specifik viktmatris från korttidsminnet i output gate.

$\mathbf{h}_{(t-1)}$  är hidden state som hanterar korttidsminnet från föregående cell.

$\mathbf{b}_o$  är en specifik biasterm i output gate.

$\sigma$  är aktiveringsfunktionen Sigmoid.

$\mathbf{y}_{(t)}$  är slutliga prediktionen.

$\tanh$  är aktiveringsfunktionen tanh

I Output gate multipliceras till sist korttidsminnet och input med vardera viktmatriser och adderas återigen med en biasterm. Summan går sedan igenom en sigmoid-funktion och  $o_t$  fås. För att sedan få en output tas det uppdaterade långtidsminnet från cell state  $C_t$  genom en tanh-funktion och multipliceras sedan elementvis med  $o_t$ . Där efter fås slutligen en output som blir det uppdaterade korttidsminnet för nästkommande LSTM-cell fram tills att den sista tidpunkten har passerat alla gates, då blir given output en slutlig prediktion.

### 3.9 Gated Recurrent Unit

GRU, eller Gated Recurrent Unit, är en variant av RNN som utvecklats för att lösa minnesproblem, samt problem med gradient vanishing och explosion som förekommer i vanliga RNN-modeller. Den introducerades av Cho et al. år 2014 [27] och delar vissa likheter med LSTM men med en något enklare arkitektur.

GRU har två grindar: en reset gate (återställningsgrind) och en update gate (uppdateringsgrind). Reset gate avgör hur mycket av den nya informationen ska användas för att uppdatera cellens tillstånd, medan update gate avgör hur mycket av den tidigare informationen som ska behållas. Genom att använda dessa två grindar kan GRU effektivt hantera långa sekvenser och memorera relevanta mönster i data.

Tack vare sin enklare struktur och liknande prestanda som LSTM har GRU blivit populär inom många tillämpningar inom maskininlärning och djupinlärning, särskilt när det gäller att modellera sekvensdata.

Vidare kan en GRU-cell uttryckas enligt:

$$\begin{aligned} \mathbf{z}_{(t)} &= \sigma(\mathbf{W}_{xz}^\top \mathbf{x}_{(t)} + \mathbf{W}_{hz}^\top \mathbf{h}_{(t-1)} + \mathbf{b}_z) \\ \mathbf{r}_{(t)} &= \sigma(\mathbf{W}_{xr}^\top \mathbf{x}_{(t)} + \mathbf{W}_{hr}^\top \mathbf{h}_{(t-1)} + \mathbf{b}_r) \\ \mathbf{g}_{(t)} &= \tanh(\mathbf{W}_{xg}^\top \mathbf{x}_{(t)} + \mathbf{W}_{hg}^\top (\mathbf{r}_{(t)} \otimes \mathbf{h}_{(t-1)}) + \mathbf{b}_g) \\ \mathbf{h}_{(t)} &= \mathbf{z}_{(t)} \otimes \mathbf{h}_{(t-1)} + (1 - \mathbf{z}_{(t)}) \otimes \mathbf{g}_{(t)} \end{aligned} \tag{3.26}$$

där

$W$  är en viktmatris i cellen

$h$  är minnet i cellen

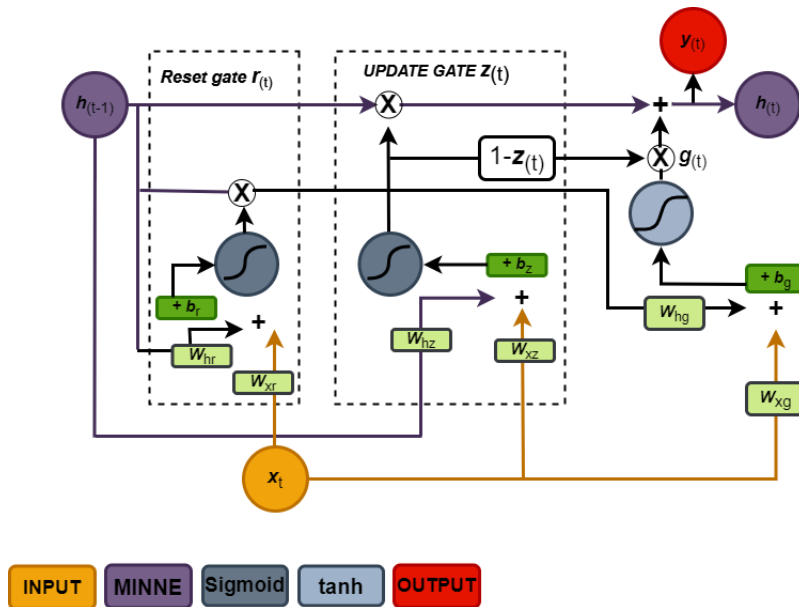
$b$  är en biasterm i i cellen

$r_{(t)}$  är reset gate

$z_{(t)}$  är update gate

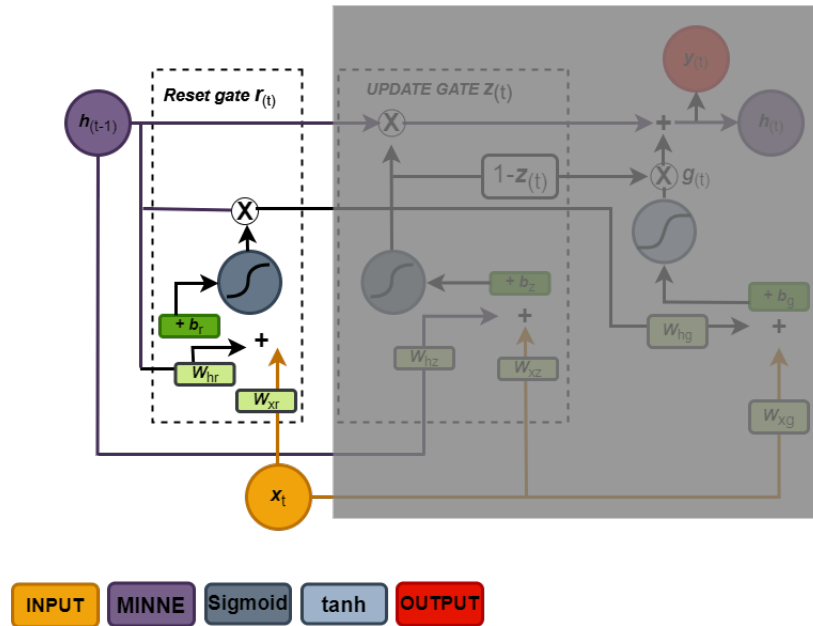
$g_{(t)}$  är sista steget innan output

$h_{(t)}$  är det nya minnet eller, för sista tidpunkten slutliga prediktionen  $y_{(t)}$



Figur 3.14: Cellstruktur GRU

I figur 3.14, inspiration från [22] syns nu hur en GRU-cell fungerar i praktiken. Strukturen påminner om hur en LSTM-cell fungerar men som nämnt tidigare är GRU mindre komplex än en LSTM-cell. Det går likt LSTM-cellen även här att förklara det som att cellen tar två inflöden av information i sig, det första är en input som skickar in det data som ska predikteras och det andra är ett hidden state  $h_{t-1}$  som bestämmer hur mycket av informationen som kommer in som ska sparas och fungerar som ett minne i cellen.



Figur 3.15: Reset gate GRU

$$\mathbf{r}_{(t)} = \sigma(\mathbf{W}_{xr}^\top \mathbf{x}_{(t)} + \mathbf{W}_{hr}^\top \mathbf{h}_{(t-1)} + \mathbf{b}_r) \quad (3.27)$$

där

$\mathbf{r}_{(t)}$  är reset gate.

$\mathbf{W}_{xr}$  är en specifik viktmatris för input i reset gate.

$\mathbf{x}_{(t)}$  är en input av tidpunkt  $t$  i data.

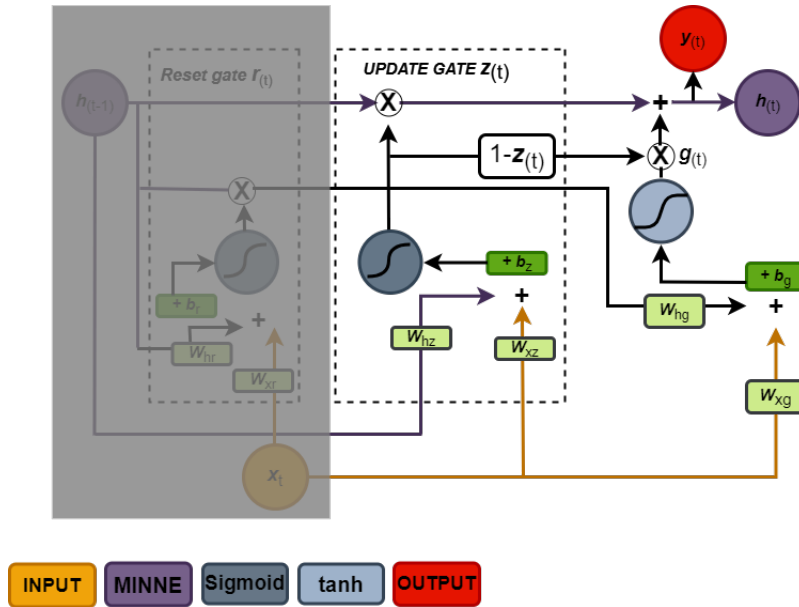
$\mathbf{W}_{hr}$  är en specifik viktmatris från minnet i reset gate.

$\mathbf{h}_{(t-1)}$  är hidden state som hanterar minnet från föregående cell.

$\mathbf{b}_r$  är en specifik biasterm i reset gate.

$\sigma$  är aktiveringsfunktionen Sigmoid.

I cellens reset gate syns hur input  $x_t$  multipliceras med en viktmatris  $w_{xr}$  och minnet i  $h_{t-1}$  multipliceras med en viktmatris  $w_{hr}$  och därefter adderas med ett bias som sedan går igenom en sigmoidfunktion innan det slutligen multipliceras elementvis med samma minne från  $h_{t-1}$  fast som här ej har transformerats på något vis innan och blir då  $r_{(t)}$ .



Figur 3.16: Update gate GRU

$$\begin{aligned}
 \mathbf{z}_{(t)} &= \sigma(\mathbf{W}_{xz}^\top \mathbf{x}_{(t)} + \mathbf{W}_{hz}^\top \mathbf{h}_{(t-1)} + \mathbf{b}_z) \\
 \mathbf{g}_{(t)} &= \tanh(\mathbf{W}_{xg}^\top \mathbf{x}_{(t)} + \mathbf{W}_{hg}^\top (\mathbf{r}_{(t)} \otimes \mathbf{h}_{(t-1)}) + \mathbf{b}_g) \\
 \mathbf{h}_{(t)} &= \mathbf{z}_{(t)} \otimes \mathbf{h}_{(t-1)} + (1 - \mathbf{z}_{(t)}) \otimes \mathbf{g}_{(t)}
 \end{aligned}
 \tag{3.28}$$

där

$\mathbf{z}_{(t)}$  är update gate.

$\mathbf{g}_{(t)}$  är kandidataktiveringsvektorn.

$\mathbf{W}_{xz}$ ,  $\mathbf{W}_{xg}$  är en specifik viktmatris för input i update gate och kandidataktiveringsvektorn.

$\mathbf{x}_{(t)}$  är en input av tidpunkt  $t$  i data.

$\mathbf{W}_{hz}$ ,  $\mathbf{W}_{hg}$  är en specifik viktmatris från minnet i update gate och kandidataktiveringsvektorn.

$\mathbf{h}_{(t-1)}$  är hidden state som hanterar minnet från föregående cell.

$\mathbf{b}_z$ ,  $\mathbf{b}_g$  är en specifik biasterm i update gate och kandidataktiveringsvektorn.

$\sigma$  aktiveringsfunktionen Sigmoid.

$\tanh$  aktiveringsfunktionen tanh

$\mathbf{h}_{(t)}$  är det nya minnet fram tills sista tidpunkten, sedan  $\mathbf{y}_{(t)}$ .

Vidare syns det i cellens update gate hur en input  $x_t$  och minnet från  $h_{t-1}$  multipliceras med respektive viktmatris  $w_{xz}$  och  $w_{hz}$  och sedan adderas med ett bias  $b_z$  innan det går igenom en sigmoid-funktion och blir  $z_{(t)}$ . Där efter sker två saker: Det första som sker är att 1 subtraheras med  $z_{(t)}$  samtidigt som  $z_{(t)}$  elementvis multipliceras med det icke-transformerade minnet från  $h_{t-1}$ .

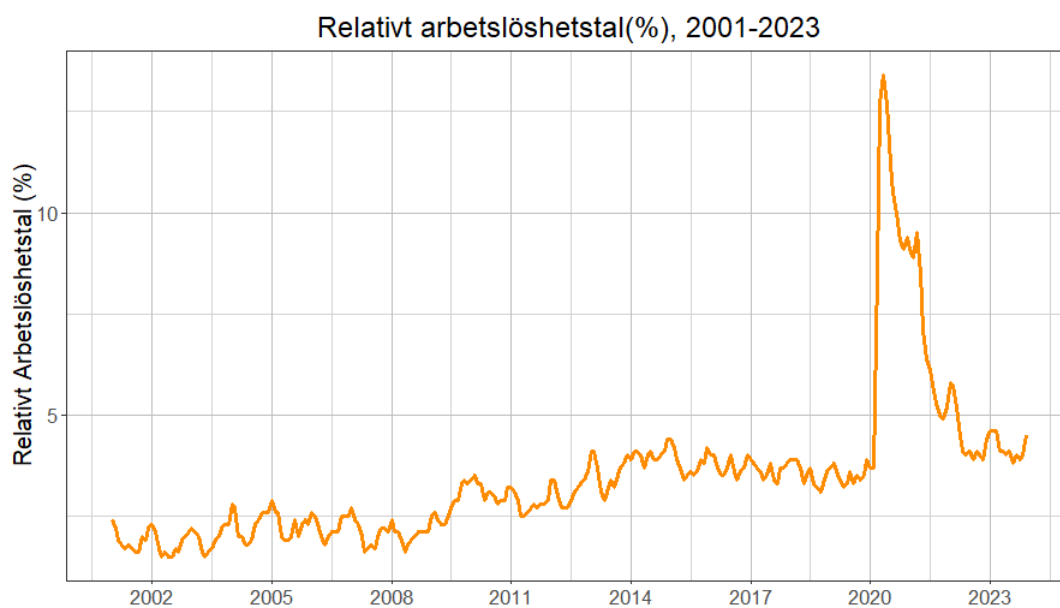
Slutligen, när  $x_t$  och  $h_{t-1}$  gått igenom reset gate och update gate ska de till sist uppdatera hidden state  $h_{t-1}$  och skapa ett nytt  $h_t$  som blir det nya minnet. Detta sker då genom att  $x_t$  och produkten av  $r_{(t)}$  och  $h_{t-1}$  multipliceras med respektive viktmatris  $w_{xg}$  och  $w_{hg}$ , där adderas de sedan ihop med ett bias  $b_g$  innan det går igenom en tanh-funktion och blir till sist en kandidataktiveringsvektor  $g_{(t)}$ . Efter detta multipliceras kandidataktiveringsvektorn elementvis med differensen  $1 - z_{(t)}$  och slutligen adderas med produkten av  $z_{(t)}$  och  $h_{t-1}$  innan det till sist blir det nya minnet  $h_t$  fram tills att sista tidpunkten i input  $x_t$  har gått igenom cellen, efter det fås en output  $y_t$  som blir den slutliga prediktionen.





## 4. Resultat

I detta kapitel kommer resultatet från totalt sex stycken modeller att redovisas där tre av modellerna har skattats på en obeskuren serie där det relativa arbetslöshetsstalet har predikterats för år 2024. De andra tre modellerna har skattats på en beskuren serie där relativa arbetslöshetsstalet predikterats för år 2019, detta för att ta hänsyn till covid-19. Som nämnt tidigare kommer detta kapitel att spegla vad prediktionerna för den relativa arbetslösheten på Åland givit för resultat, där totalt 34 variabler från åtta makroekonomiska områden använts, vilka var Relativt arbetslöshetsstal, Ungdomsarbetslöshetsstal, Lediga platser / arbetslösa, Konsumentprisindex, Inresande till Åland, Näringslivets löneutbetalningar, Näringslivets omsättning samt Antal covid-19 fall.



Källa: Ålands statistik- och utredningsbyrå (ÅSUB)

Figur 4.1: Relativt arbetslöshetsstal(%), 2001-2023

## 4.1 Implementering i R och Python

Val av fönsterstorlek för datamaterialet samt modellskattningarna har genomförts i datorspråket Python där paketet Keras har använts för att utföra uppgiften, vidare har även early stopping använts och ställts in på att gå tillbaka tre epoker efter att inget bättre MSE i valideringen observerats. Vidare har sedan modellerna visualiserats i datorspråket R där paketet ggplot2 använts. I tabell 4.1 nedan visas de hyperparametrar som producerat den bästa modellen för samtliga sex modeller.

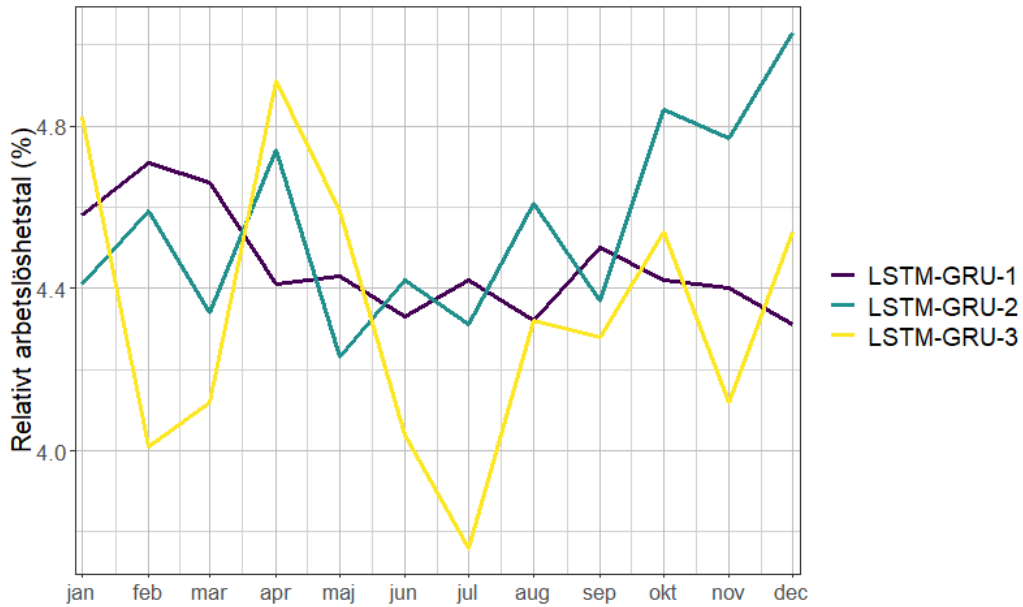
Tabell 4.1: Modellinformation

Modell	Osäkerhet	Early stopping	Hyperparametrar						MSE <sub>V</sub>	GNLL <sub>V</sub>	MSE <sub>T</sub>
			Fönsterstorlek	Neuroner	Learning rate	Gömda lager	Batchstorlek	Dropout (p)			
LSTM-GRU 1	Epistemisk	27	24	50/50	0.0001	2	1	0.3	1.813	-	-
LSTM-GRU 2	Aleatorisk	3	24	80/80	0.001	2	1	-	1.265	1.119	-
LSTM-GRU 3	Epistemisk + Aleatorisk	13	24	88/88	0.0001	3	1	0.3	1.499	1.406	-
LSTM-GRU 4	Epistemisk	15	24	256/128	0.0001	2	1	0.3	0.104*	-	0.040
LSTM-GRU 5	Aleatorisk	47	24	80/80	0.0001	2	1	-	0.133*	0.574	0.055
LSTM-GRU 6	Epistemisk + Aleatorisk	54	24	80/80	0.0001	2	1	0.3	0.319*	1.398	0.047

I tabell 4.1 redovisas det vilka parametrar som inkluderats för varje modell samt utvärderingsmått. Modell 1, 2 och 3 avser alla en modell med LSTM-GRU som skattats på en obesuren serie där det relativa arbetslöshetsmättet predikterats för år 2024. Notera att för dessa tre så har inte Test-MSE kunnat beräknats då hela datamängden använts för träning av modellen.

Modell 4, 5 och 6 avser alla en modell med LSTM-GRU men som här skattats på en besuren serie där det relativa arbetslöshetsmättet predikterats för år 2019. Notera att Validerings-MSE här har använts för att kunna jämföra mot de tre modeller som skattats för den obesurna serien och de tre modellerna som skattats för den besurna serien. Validerings-MSE har dock inte använts för att minimera förlustfunktionen för prediktionerna för den besurna serien utan där har Validerings-GNLL använts.

Prediktion Relativt arbetslöshetstal 2024 för respektive modell

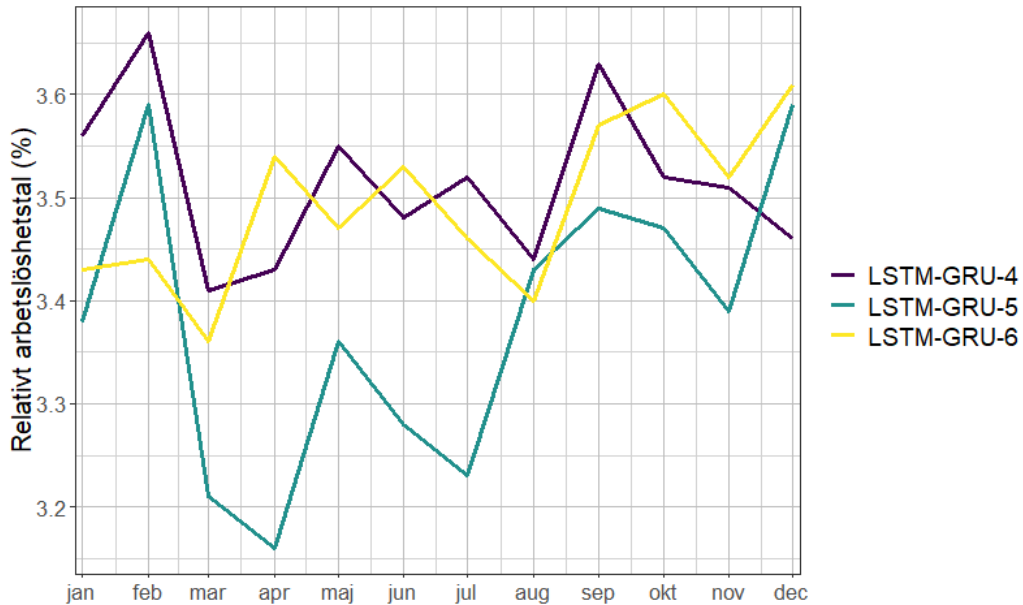


Källa: Egna prediktioner på data från ÅSUB

Figur 4.2: Prediktion av Relativt arbetslöshetstal 2024 för respektive modell

I figur 4.2 syns prediktionen för det relativa arbetslöshetstalet på Åland 2024 för LSTM-GRU modell 1-3. Det går att se att prediktionerna är relativt lika och har relativt liknande mönster. Modell 1 är den med minst fluktueringar medan modell 3 har kraftiga fluktueringar upp och ned.

Prediktion Relativt arbetslöshetstal 2019 för respektive modell



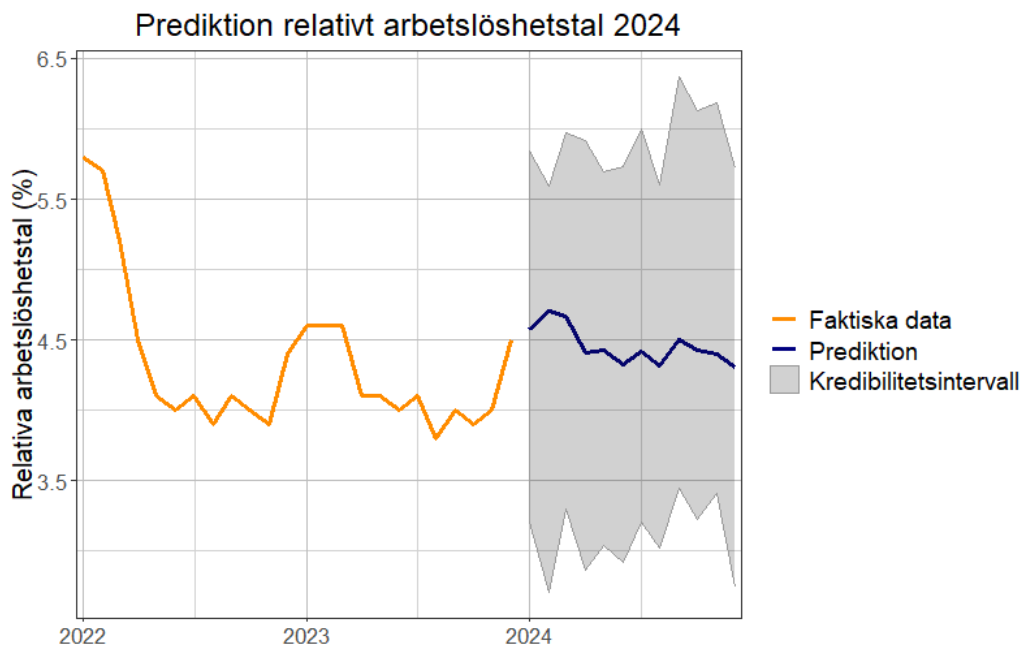
Källa: Egna prediktioner på data från ÅSUB

Figur 4.3: Prediktion av Relativt arbetslöshetstal 2019 för respektive modell

I figur 4.3 syns prediktionen för det relativa arbetslöshetstalet på Åland 2024 för LSTM-GRU modell 4-6. Det går att se att prediktionerna även här är relativt lika och har relativt liknande mönster. Modell 4 och modell 6 följs åt relativt bra från mars och framåt medan modell 5 har ett liknande mönster som modell 4 men något större fluktueringar och lägre värden för relativa arbetslöshetstalet.

## 4.2 LSTM-GRU 1

LSTM-GRU 1 ämnar att ta en epistemisk osäkerhet i beaktande. I tabell 1 går det att se all information kring modellen och dess hyperparametrar. De två lager som används är ett LSTM- och ett GRU-lager med 50 neuroner i sig vardera. Efter att modellen tränats används de sista 24 månaderna (januari 2022 - december 2023) för att prediktera nästkommande 12 månader (år 2024).



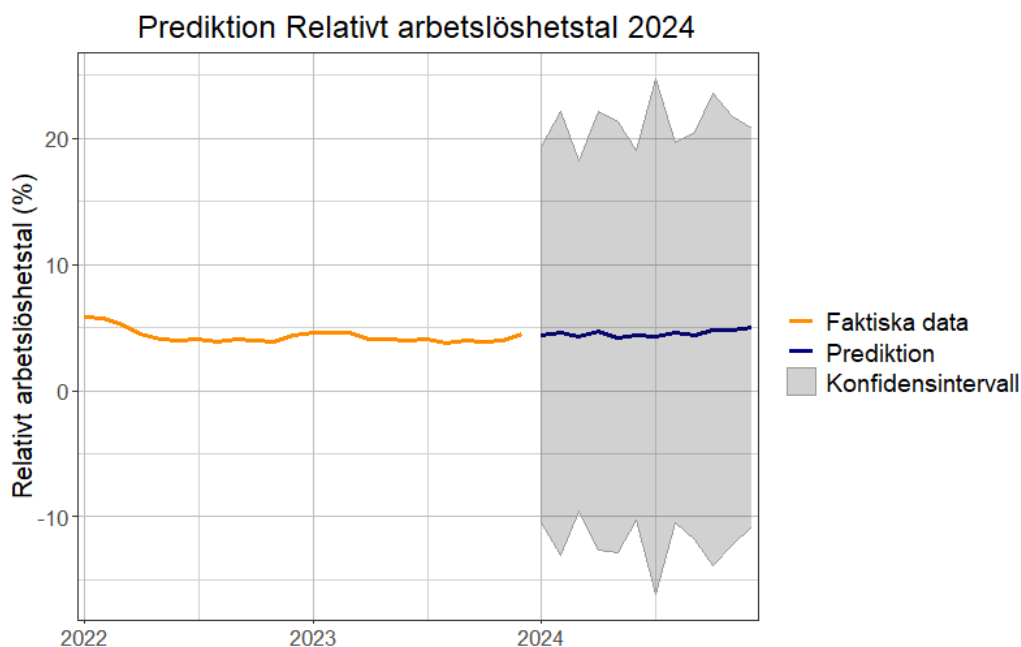
Källa: Egna prediktioner på ÅSUBs data

Figur 4.4: LSTM-GRU med epistemisk osäkerhet

I figur 4.4 syns nu prediktionerna för de nästkommande 12 månaderna. Då MC-Dropout har använts har modellen skapat 500 prediktioner, sedan har alla prediktioner aggregerats och sedan har ett medelvärde för varje tidpunkt beräknats. Eftersom de 500 prediktionerna enligt teorin ger en fördelning beräknas även 95% kredibilitetsintervall för prediktionerna som syns som det gråmarkerade området i grafen. Kredibilitetsintervallen är mellan ca. 2.8 och 6.2 procentenheter. Arbetslösheten predikterades till ett genomsnitt på 4.45% för år 2024.

## 4.3 LSTM-GRU 2

LSTM-GRU 2 ämnar att ta en aleatorisk osäkerhet i beaktande. I tabell 1 går det att se all information kring modellen och dess hyperparametrar där både ett LSTM- och ett GRU-lager använts med 50 neuroner i sig vardera. Efter att modellen tränats används de sista 24 månaderna (januari 2022 - december 2023) för att prediktera nästkommande 12 månader (år 2024).



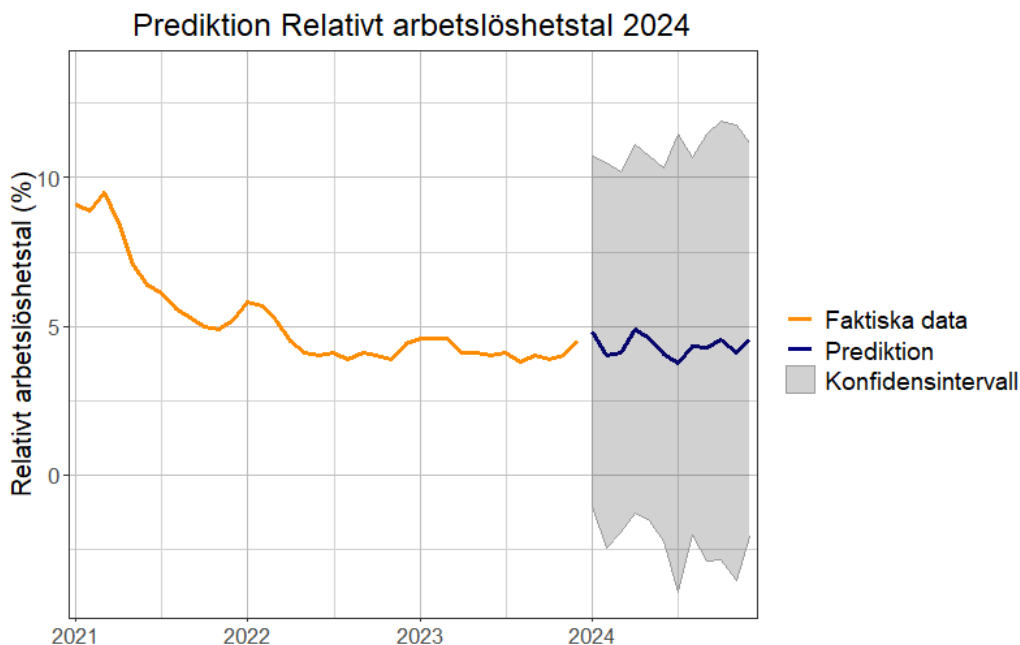
Källa: Egna prediktioner på data från ÅSUB

Figur 4.5: LSTM-GRU med aleatorisk osäkerhet

I figur 4.5 syns nu prediktionerna för de nästkommande 12 månaderna. Därefter har de skattade log-varianserna använts och ett 95 % konfidensintervall har beräknats som syns i det gråmarkerade området. Intervallen sträcker sig från ca -16.13 till 25.75 procentenheter. Arbetslösheten predikterades till ett genomsnitt på 4.55 % för år 2024. Resultatet är inte att klassa som rimligt, detta då osäkerheten här säger att arbetslösheten kan ligga på vilken nivå som helst men att den även kan vara negativ, vilket förstås inte är möjligt.

## 4.4 LSTM-GRU 3

LSTM-GRU 3 ämnar att ta en epistemisk och aleatorisk osäkerhet i beaktande. I tabell 1 går det att se all information kring modellen och dess hyperparametrar där två LSTM- och ett GRU-lager använts med 88 stycken neuroner i sig vardera. Efter att modellen tränats används de sista 24 månaderna (januari 2022 - december 2023) för att prediktera nästkommande 12 månader (år 2024).

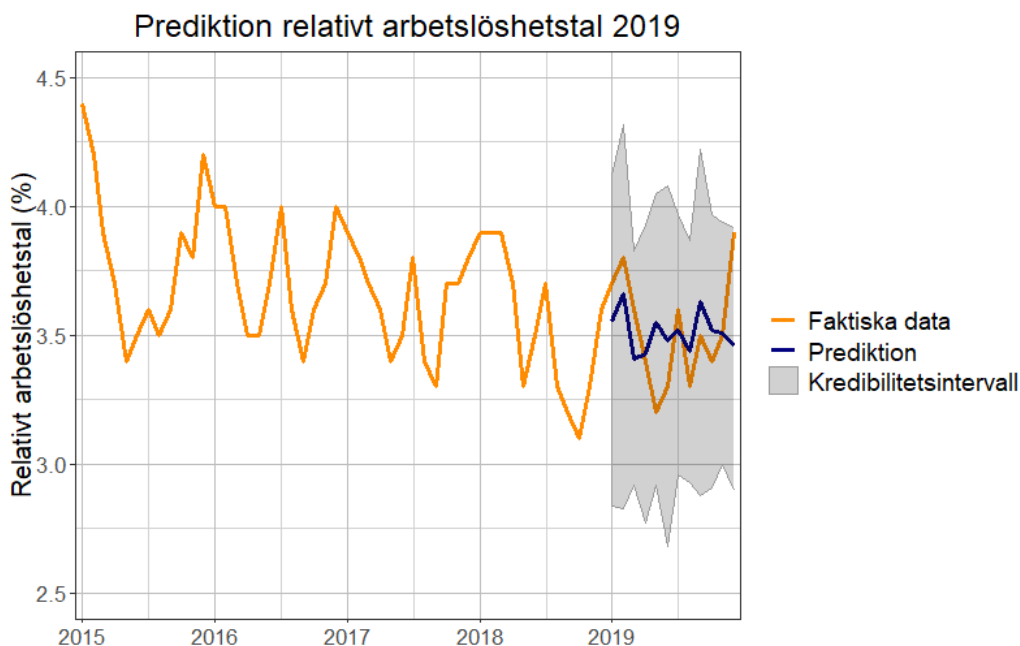


Figur 4.6: LSTM-GRU med epistemisk och aleatorisk osäkerhet

I figur 4.6 syns nu prediktionerna för de nästkommande 12 månaderna. Därefter har MC-Dropout och de skattade logvarianserna använts och modellen har tillkallats 50 gånger för att ta fram den prediktiva osäkerheten och vidare har ett 95 % konfidensintervall skattats som syns i det gråmarkerade området där intervallen sträcker sig från ca -3.65 till 12.56 procentenheter. Arbetslösheten predikterades till ett genomsnitt på 4.34 % för år 2024. Resultatet av att använda en kombination mellan epistemisk och aleatorisk osäkerhet har här visat sig generera en bättre prediktion än att bara använda en aleatorisk, notera dock att resultatet här inte heller är att klassa som rimligt, detta med samma grund som för LSTM-GRU 2.

## 4.5 LSTM-GRU 4

LSTM-GRU 4 ämnar att ta en epistemisk osäkerhet i beaktande men här för det beskurna datamaterialet. I tabell 1 går det att se all information kring modellen och dess hyperparametrar. De två lager som används är ett LSTM-lager med 256 neuroner i sig och ett GRU-lager med 128 neuroner i sig, totalt har modellen 450 732 parametrar. Efter att modellen tränats används de sista 24 månaderna (januari 2017 - december 2018) för att prediktera nästkommande 12 månader (år 2019).



Källa: Egna prediktioner på ÅSUBS data

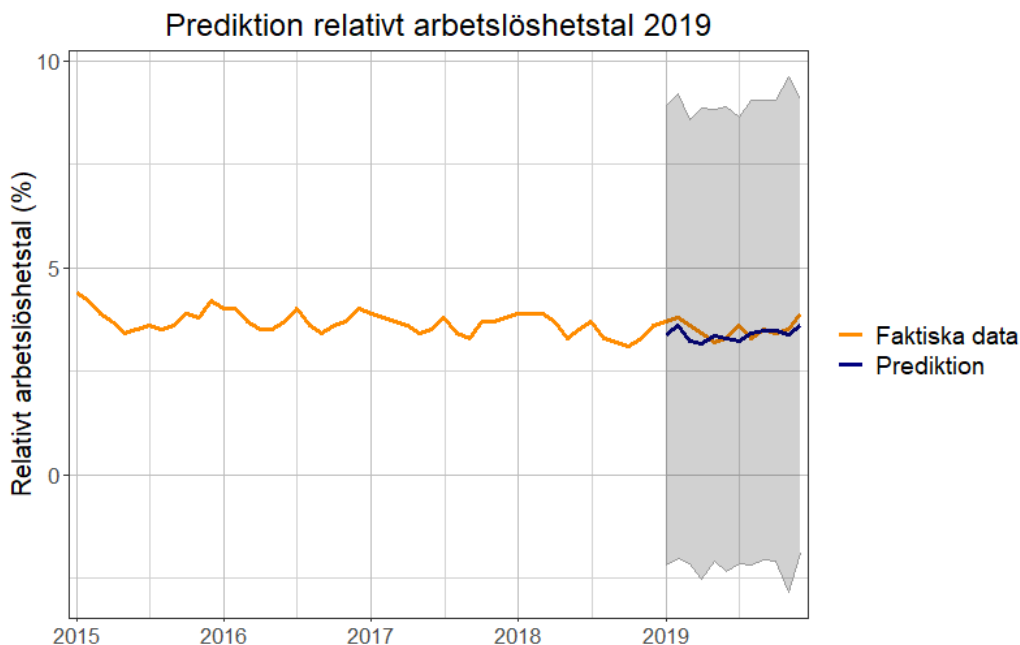
Figur 4.7: LSTM-GRU med epistemisk osäkerhet, beskuret datamaterial

I figur 4.7 syns nu prediktionerna för de nästkommande 12 månaderna. Likt tidigare skapades även här 500 prediktioner. Eftersom det finns tillgång till de riktiga värdena för 2019 har de lagts in i grafen. Även här skapades sedan ett 95 % kredibilitetsintervall för att visualisera osäkerheten. Det syns att prediktionerna är något mer tillplattade jämfört med de riktiga värdena. Prediktionerna för det relativa arbetslöshetstalet visar att det kommer vara ca. 3.51 % medan de riktiga värdena för 2019 redovisade ett relativt arbetslöshetstal på 3.52 %. Kredibilitetsintervallen går mellan ca 2.68 och 4.3 procentenheter. Dessa intervall är något smalare än de intervall som observerades i LSTM-GRU 1.



## 4.6 LSTM-GRU 5

LSTM-GRU 5 ämnar att ta en aleatorisk osäkerhet i beaktande men här för det beskurna datamaterialet. I tabell 1 går det att se all information kring modellen och dess hyperparametrar där både ett LSTM- och ett GRU-lager använts med 80 stycken neuroner i sig vardera. Efter modellens träning används de sista 24 månaderna (januari 2017 - december 2018) för att prediktera nästkommande 12 månader (år 2019).



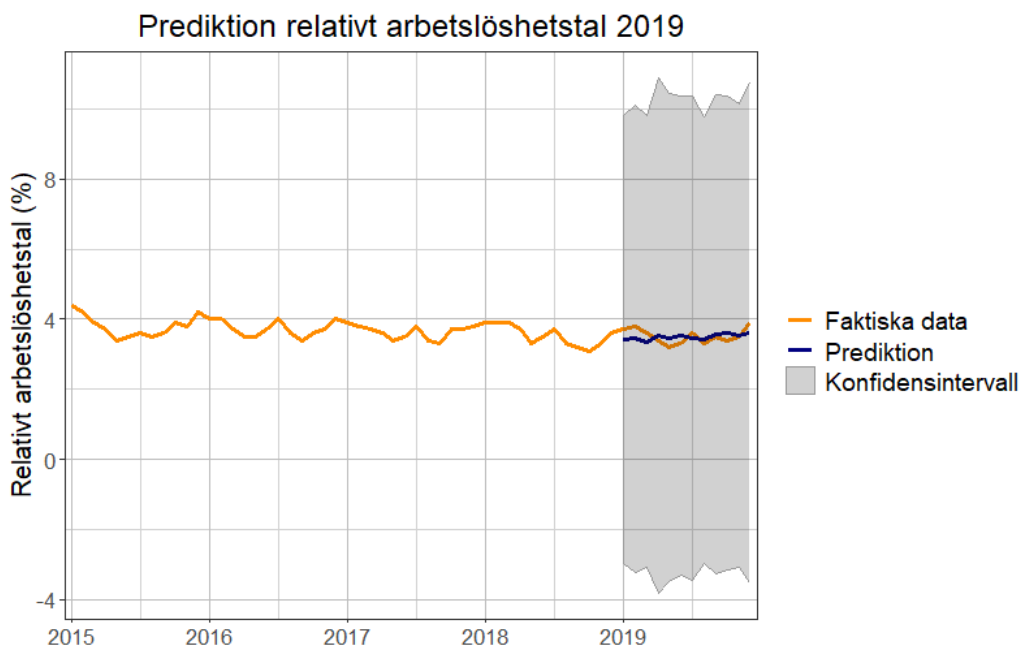
Källa: Egna prediktioner på ÅSUBs data

Figur 4.8: LSTM-GRU med aleatorisk osäkerhet, beskuret datamaterial

I figur 4.8 syns nu prediktionerna för de nästkommande 12 månaderna. Därefter har de skattade log-varianserna, sedan har ett 95 % konfidensintervall beräknats där intervallen sträcker sig från ca -2.84 till 9.61 procentenheter. Arbetslösheten predikterades till ett genomsnitt på 3.38 % för år 2019. Prediktionen här genererade ett bättre resultat än för den där serien behölls obeskuren men med samma grund som för tidigare prediktioner är inte detta resultat heller att betrakta som rimligt.

## 4.7 LSTM-GRU 6

LSTM-GRU 6 ämnar att ta en epistemisk och aleatorisk osäkerhet i beaktande men här för det beskurna datamaterialet. I tabell 1 går det att se all information kring modellen och dess hyperparametrar där två LSTM- och ett GRU-lager använts med 80 stycken neuroner i sig vardera. Efter att modellen tränats används de sista 24 månaderna (januari 2017 - december 2018) för att prediktera nästkommande 12 månader (år 2019).



Källa: Egna prediktioner på ÅSUBs data

Figur 4.9: LSTM-GRU med epistemisk och aleatorisk osäkerhet, beskuret datamaterial

I figur 4.9 syns nu prediktionerna för de nästkommande 12 månaderna. Därefter har de skattade logvarianserna och MC-Dropout använts och sedan har 50 prediktioner skapats, därefter har ett 95 % konfidensintervall beräknats där intervallen sträcker sig från ca -3.8 till 10.9 procentenheter. Arbetslösheten predikterades till ett genomsnitt på 3.49 % för år 2019. Prediktionerna visade här en något mindre osäkerhet än vad LSTM-GRU-3 gjorde men faktum kvarstår även här att resultatet visar på en så pass stor osäkerhet att det inte heller är att klassa som rimligt.

## 5. Diskussion/Analys

Under följande kapitel kommer olika delar av uppsatsen att lyftas samt diskuteras, här kommer även förslag till framtida forskning att ges.

### 5.1 Prediktioner

Vad det gäller resultatet av modellskattningarna så hade bättre prediktioner önskats vad det gäller de som tagit en aleatorisk och en kombination av aleatorisk och epistemisk osäkerhet i beaktande. Då konfidensintervallen för dessa är väldigt breda och dessutom täcker 0 så kan inget annat än ett fastställande att säkerheten kring att prediktionerna inte är tillförlitliga göras. Det är i sin tur något som går hand i hand med att det är just en aleatorisk osäkerhet som har tagits i beaktande, då den aleatoriska osäkerheten lämpar sig bättre för typer av data där antalet observationer i datamängden är betydligt större än vad som funnits tillgång till, vilket då talar för att det kanske inte var lämpligt att använda en aleatorisk osäkerhet i modellen. [25]

Modeller där en epistemisk osäkerhet har tagits i beaktande har däremot genererat goda resultat, där syns mycket mindre intervall och som dessutom inte täcker 0. Detta talar för att de prediktioner som modellerna genererat är tillförlitliga och där LSTM-GRU-4 som predikerade arbetslösheten för 2019 var den som gav bäst resultat sett till att dess kredibilitetsintervall var smalare än för LSTM-GRU 1. Validerings-MSE var som lägst vid prediktionen för 2019 och det facto att det riktiga genomsnittliga värdet för det relativa arbetslöshetstalet 2019 var 3.52 % och LSTM-GRU-4 genererade en genomsnittlig prediktion på arbetslösheten på 3.51 %. Därav, med stöd från tidigare studier går det att klargöra att LSTM-GRU med MC-Dropout är lämpligare att använda än modeller som använde en aleatorisk osäkerhet. [25]

Vid jämförelse utav ÅSUBs prognoser för 2019 hittades tre olika prediktioner. Våren 2018 predikterades det relativa arbetslöshetstalet att komma att ligga kring 3.7 % 2019. [28] Våren 2019 var prediktionen för hela 2019 3.5 %. [29] Denna prediktionen hade de gjort med första kvartalets data som underlag till prediktionen också. Sista funna prediktionen var gjord hösten 2019, och då var prediktionen 3.4 %. [30] De riktiga värdet för 2019 är, som tidigare nämnt, 3.52 %. LSTM-GRU 4 är den modell som gett bäst resultat för 2019 med en skattning på 3.51 %. Jämförs alla dessa prognoser syns det snabbt att LSTM-GRU 4 är den modell som ligger närmast de riktiga värdena medan ÅSUBs prognoser ligger aningen längre ifrån.

Under arbetets gång har det relativa arbetslöshetstalet för 2024 börjat publicerats [11], vilket leder till att man kan jämföra detta mot de olika modellernas prediktioner. Vid skrivandet har det relativa arbetslöshetstalet fram till april publicerats. De riktiga värden är följande; januari: 4.6%, februari: 4.7%, mars: 4.6% och april: 4.4%. Jämför man detta med våra skattade modeller kan man se att LSTM-GRU-1 följer detta relativt bra

med punktprediktionerna: januari: 4.58%, februari: 4.71%, mars: 4.66% och april: 4.44%. Vid jämförelse utav ÅSUBs modell blir det dock något svårare då den endast är på en ettårsbasis istället för månadsvis som har gjorts i detta arbete. Dock är deras prediktion för hela 2024 4.3 % [31] jämfört med vår model LSTM-GRU-1 som har 4.45%.

Det är också värt att tillägga att tidigare studier menat på att en kombination av aleatorisk och epistemisk osäkerhet inte heller är särskilt lämpligt då det finns en risk för att de tenderar att jobba emot varandra vilket då inte heller genererar särskilt goda resultat som ju här beskådats, valet att kombinera aleatorisk och epistemisk osäkerhet grundade sig i att då den epistemiska visade så pass bra resultat så föddes tanken att MC-Dropout möjligen kunde hjälpa till att få bättre resultat på skattningarna, men som presenterat i resultatkapitlet, uteblev de resultaten. [25]

## 5.2 Studiens jämförbarhet

Huruvida denna studie är helt jämförbar med ÅSUBs rapporter kring arbetslösheten går också att diskutera, bland annat sett utifrån brist i tillgång till alla ÅSUBs variabler på grund av konfidentiella skäl. Då information om vilken modell de använde när de skapade sina prediktioner också var bristande kunde varken deras resultat reproduceras eller användas föra att inkludera alla deras variabler i LSTM-GRU, vilket hade varit det optimala för att kunna skapa så bra och jämförbara resultat som möjligt.

Angående Näringslivets omsättning och Näringslivets löneutbetalningar bör det nämnas att de uppgifterna för de senaste två åren i vardera datamängd var preliminära vilket kanske inte heller är helt lämpligt att inkludera i modellen, men då mängden data som sagt var i det minsta laget så fick ett beslut om att använda dessa ändå tas, dels för att det hade inneburit en för stor dataförlust att ta bort två år redan som det är men också för att om det ej hade gjorts så hade slutet av datamängden hamnat mitt i covid-19-pandemin, vilket inte heller hade gynnat prediktionerna då coronaspiken var alldeles för stor. [16] [15]

## 5.3 Modellutvärdering

För att kunna jämföra alla modeller sinsemellan hade test-MSE helst använts som utvärderingsmått. Det var dock inte genomförbart då det hade inneburit en betydande problematik för den obeskrivna serien. Detta eftersom det hade inneburit en beskärning av en del av data som testdata, vilket i sin tur hade medfört att träningsdata hade slutat mitt i coronapandemin då 10 % som testmängd inneburit ungefär 28 månader. Därför togs beslutet att använda ett validerings-MSE, som kunde beräknas för alla modeller även om det inte var det lämpligaste tillvägagångssättet. Däremot ansågs det som en god idé att använda test-MSE för de beskrivna modellerna eftersom att det var genomförbart där, varpå det syntes att LSTM-GRU 4 även där var den bästa modellen.

Med stöd från tidigare forskning kring maskininlärningsmetoder för tidsseriedata har Modellerna som skapats alla använt LSTM/GRU som struktur. Under arbetets gång testades modeller med enbart LSTM, enbart med GRU och kombinationer mellan de sinsemellan. Olika tester av mängder lager av LSTM och GRU, olika antal neuroner i varje och andra hyperparametrar testades också där en kombination mellan dessa visade sig vara den bästa strukturen att använda. Användes endast LSTM-lager överanpassade sig modellerna redan vid första epoken medan om bara GRU användes så blev resultaten undermåliga vilket syntes genom valideringsmse som blev väldigt högt och alltid högre än valideringsmse för LSTM-modellen. Att använda strukturen

LSTM-GRU har under arbetets gång varit en utmaning att förstå sig på hur de tar sig an tidsserieproblem då vi aldrig tidigare bekantas oss med sådana metoder innan under vår studietid. Däremot ansåg vi fortfarande i början av uppsatsen att det skulle vara det bästa sättet att ta sig an uppgiften på och då vi hade tidigare kännedom kring neurala nätverk så ansåg vi också att utmaningen fortfarande var genomförbar under tiden vi hade att tillgå. Under arbetets gång så har vi också successivt lärt oss konceptet bakom metoderna och har nu landat i att det trots allt var den bästa vägen att gå. [3][32]

För samtliga modeller har ett flertal olika kombinationer av alla hyperparametrar testats och optimerats konsekvent och systematiskt genom arbetets gång. För varje modell började vi med att implementera ett LSTM- och ett GRU-lager, där neuroner, learnig rate dropout sattes till rimliga värden. Därefter tränades modellen och ett resultat gavs som sedan kunde analyseras och utvärderas, därefter valde vi att justera antal neuroner samt learning rate och andel dropout i de befintliga lagrena och tränade om igen. Efter att den processen genomförts några gånger lade vi till ytterligare ett LSTM- och GRU-lager, fortsatte processen som nämnd ovan och därefter testade att ta bort antingen ett av, eller de båda nyligen tillagda lagrena, genererade det ett bättre resultat så valde vi att behålla strukturen som den var och sedan optimera därifrån. När vi sedan kommit fram till ett resultat som vi utefter rimliga gränser kunde känna oss nöjda med så valde vi att ta med den strukturen till nästa modell som en grund. Därifrån upprepades ovan nämnd process fram tills att den bästa modellen identifierats. Under arbetets gång testades även automatiska sökningar efter hyperparametrar, såsom bland annat grid search men som inte gav något bra resultat vilket i sin tur ledde till att vi själva fick hitta på vårt egna tillvägagångssätt som resulterade i den process som beskrivet ovan. Huruvida automatiska sökningar är att föredra går att diskutera utifrån fler aspekter, det sparar ju mycket tid att göra det automatiskt och det kan även underlätta för diverse beräkningar. Däremot bygger ju en automatisk lösning på att resultatet också blir bra, vilket ju inte var fallet för oss.

Vad det gäller valet av att använda modeller som tog en epistemisk osäkerhet i beaktande så stöttes det på problem med MC-Dropout då teorin bakom metoden var mycket komplex och över den nivå som detta arbete bör ligga under. Konceptuellt är MC-Dropout inte komplext och att implementera det i modellerna är inte heller något problem utan här föll det på att beviset för att metoden fungerar var det svåra men som löstes genom att beskriva metoden konceptuellt samt källhänvisa till dess originalartikel [24].

Effekten av covid-19 visade sig också vara betydande under arbetet där det observerades att skattningarna generellt blev bättre när de ej behövde ta hänsyn till en så stor spik som ju pandemin faktiskt orsakade i data. Då vi idag lever i 2024 och inte kan radera covid-19 ur världen utan bara får leva med att det faktiskt hände och att dess påverkan drabbat oss alla så är det något som man kommer att behöva ta hänsyn till. I detta fall visade det ju sig att det uppstår problem när pandemin är där vilket ställer frågor kring modellernas generaliserbarhet. Vad vi däremot vill nämna är att vi tror att detta är ett problem som uppstår hur man gör och att det må vara en utmaning för framtiden att ta itu med världsomvälvande händelser i historien när man predikterar framtiden. Uppsatsen syftade ju till att undersöka ifall maskininlärningsmetoder var lämpliga att använda för att prediktera arbetslöshet, vilket resultatet visat att det är givet att händelser som covid-19 hanteras på samma sätt som i denna uppsats.

## 5.4 Framtida forskning

I samma studie som presenterade GRU presenterades även en typ av nätverk som kallas ”encoder-decoder” som är en typ av arkitektur på neurala nätverk som ofta används för denna typ av problem då det finns tillgång

till insekvenser men målet är att få utsekvenser. Nätverket består av två delar: en encoder och en decoder. Encodern ansvarar för att processera insekvenser för att skapa kontextvektorer. Detta är vektorer som bär med sig de viktigaste kontexten kring till exempel trender, säsongsvariation och annat. Dessa vektorer skickas sedan till en decoder i nätverket som använder vektorerna för att generera en utsekvens. Denna typ av nätverk används ofta vid textbaserade problem, till exempel vid översättning, generering av text. En studie som har testat detta på tidsseriedata är en studie av Lingxue Zhu [32], där de tillsammans med företaget UBER använt ett encoder-decodernätverk med LSTM och MC-Dropout. Med denna metod lyckades de uppfylla målet med studien som var att med större säkerhet kunna upptäcka avikelser för företaget. Det är något som är värt att ta upp för vidare forskning, givet att tid och resurser finns tillgängliga då detta är en än mer avancerad metod än LSTM-GRU. Vidare medför ett encoder-decodernätverk ett behov av större datamängder då den är såpass komplex vilket också var ett argument för att inte använda av den metoden under detta arbete, notera även att storleken på det data som använts i denna uppsats ligger på gränsen till att vara lämpligt för LSTM-GRU.

En ytterligare utökning på detta som snabbt stöttes på och som även visats ge bra resultat är Transformers [33]. Då transformers är än mer komplex än både LSTM-GRU och "encoder-decoder" så valdes det att inte gå vidare med den som metod, däremot anses det att det kan vara av intresse att veta om detta. Vad som ytterligare kan tas med till vidare forskning är att testa modellerna på andra nordiska länder och observera hur den predikterar arbetslösheten. Detta var något som ville göras under uppsatsen men då tiden inte räckte till fick det bli avgränsning till endast Åland. Detta arbete kan även komma att bli bättre i framtiden då flera år och månader av arbetslöshet samlats in och publicerats. Detta på grund av att en större testmängd då kan användas, vare sig en beskuren eller obeskuren serie används då arbetslöshetsnivåerna rimligtvis bör ligga mer jämnt och inte heller ha en coronaspik i närtid. Det kan också vara intressant att se om ett beaktande av en aleatorisk osäkerhet lämpar sig bättre då än vad det gjort nu.

## 6. Slutsatser

Syftet med uppsatsen var att undersöka lämpligheten i att låta ett neuralt nätverk med en hybrid av LSTM och GRU prediktera arbetslösheten på Åland. De två bästa resultaten blev en modell med LSTM-GRU där en epistemisk osäkerhet togs i beaktande och där den ena modellen predikterade arbetslösheten för 2024 och den andra arbetslösheten för 2019, den bästa modellen var den som predikterade för år 2019, då den klipptes innan covid-19-pandemin ägde rum och således inte behövde ta det i beaktande vid prediktionen.

Frågeställningen för arbetet löd enligt följande:

Är det lämpligt att använda sig av maskininlärning vid prognos av arbetslöshet?

- Ja, det är lämpligt att använda sig av maskinlärning vid prognos av arbetslöshet, givet att ingen världsomvälvande avvikelse sker, såsom covid-19.





## Litteraturförteckning

- [1] SOU. *Vad är officiell statistik? en översyn av statistiksystemet och SCB: slutbetänkande*. Slutbetänkande av Statistikutredningen 2012. Fritze, Stockholm, 2012. ISBN 978-91-38-23847-9.
- [2] Jennifer Hobbins. *Samhälle, individ och ansvar : En studie om synen på arbetslöshet*. PhD Thesis, Karlstad University, Karlstad Business School, 2016. Backup Publisher: Karlstad University, Karlstad Business School ISBN: 978-91-7063-691-2 ISSN: 1403-8099 Issue: 2016:12 Series: Karlstad University Studies.
- [3] Mustafa Yurtsever. Unemployment rate forecasting: LSTM-GRU hybrid approach. *Journal for Labour Market Research*, 57(1):18, June 2023. ISSN 2510-5027. doi: 10.1186/s12651-023-00345-8. URL <https://labourmarketresearch.springeropen.com/articles/10.1186/s12651-023-00345-8>.
- [4] Sourav Kundu and Rajshekhar Singhania. Forecasting the United States Unemployment Rate by Using Recurrent Neural Networks with Google Trends Data. *International Journal of Trade, Economics and Finance*, 11(6):135–140, December 2020. ISSN 2010023X. doi: 10.18178/ijtef.2020.11.6.679. URL <http://www.ijtef.org/index.php?m=content&c=index&a=show&catid=110&id=1047>.
- [5] Rudzani Mulaudzi and Ritesh Ajoodha. Application of Deep Learning to Forecast the South African Unemployment Rate: A Multivariate Approach. In *2020 IEEE Asia-Pacific Conference on Computer Science and Data Engineering (CSDE)*, pages 1–6, Gold Coast, Australia, December 2020. IEEE. ISBN 978-1-66541-974-1. doi: 10.1109/CSDE50874.2020.9411581. URL <https://ieeexplore.ieee.org/document/9411581/>.
- [6] M Fajar, O R Prasetyo, and S Nonalisa. Forecasting Unemployment Rate in the Time of COVID-19 Pandemic Using Google Trends Data (Case of Indonesia). *International Journal of Scientific Research in Multidisciplinary Studies*, December 2020.
- [7] ÅSUB. Dataskydd och behandling av personuppgifter (personuppgiftspolicy) | Ålands statistik- och utredningsbyrå, 2024. URL <https://www.asub.ax/sv/om-asub/dataskydd-och-behandling-personuppgifter-personuppgiftspolicy#:~:text=ÅSUB%3As%20interna%20principer%20får%20dataskydd&text=Alla%20som%20hår%20till%20ÅSUBs,%2F9%20fåreskrivs%20som%20sekretessbelagda>.
- [8] Zhi-Hua Zhou. Machine learning challenges and impact: an interview with Thomas Dietterich. *National Science Review*, 5(1):54–58, May 2017. ISSN 2095-5138. doi: 10.1093/nsr/nwx045. URL <https://doi.org/10.1093/nsr/nwx045>.
- [9] ÅSUB. ÅSUBs statistikdatabaser (PxWeb), 2024. URL [https://pxweb.asub.ax/PXWeb/pxweb/sv/Statistik/Statistik\\_BE](https://pxweb.asub.ax/PXWeb/pxweb/sv/Statistik/Statistik_BE).

- [10] Terveiden ja hyvinvoinnin laitos (THL). Bekräftade fall av corona i Finland (COVID-19), December 2023. URL <https://thl.fi/sv/statistik-och-data/material-och-tjanster/oppna-data/bekraftade-fall-av-corona-i-finland-covid-19->.
- [11] ÅSUB. Arbetslösa arbetssökande efter månad, kön och arbetslöshetsmått, 2024. URL [https://pxweb.asub.ax/PXWeb/pxweb/sv/Statistik/Statistik\\_\\_AR\\_\\_Arbetsl%C3%B6shet/AR001.px/](https://pxweb.asub.ax/PXWeb/pxweb/sv/Statistik/Statistik__AR__Arbetsl%C3%B6shet/AR001.px/).
- [12] ÅSUB. Beskrivning av Arbetslöshetsstatistiken, 2024. URL <https://www.asub.ax/sv/om-asub/dataskydd-och-behandling-personuppgifter-personuppgiftspolicy#:~:text=ÅSUB%3As%20interna%20principer%20fÅur%20dataskydd&text=Alla%20som%20hÅur%20till%20ÅSUBs,%2F9%20fÅureskrivs%20som%20sekretessbelagda.>
- [13] ÅSUB. Konsumentprisindex för Åland, procentuella förändringar under tolv månadersperioder (inflation) efter år och månad, 2024. URL [https://pxweb.asub.ax/PXWeb/pxweb/sv/Statistik/Statistik\\_\\_KO/KO000.px/](https://pxweb.asub.ax/PXWeb/pxweb/sv/Statistik/Statistik__KO/KO000.px/).
- [14] ÅSUB. Inresande till Åland 1994-2024 efter år, månad och avreseland och transportmedel, 2024. URL [https://pxweb.asub.ax/PXWeb/pxweb/sv/Statistik/Statistik\\_\\_TU\\_\\_Inresande/TU005.px/](https://pxweb.asub.ax/PXWeb/pxweb/sv/Statistik/Statistik__TU__Inresande/TU005.px/).
- [15] ÅSUB. Näringslivets omsättning (miljoner euro) efter år, månad och bransch, 2024. URL [https://pxweb.asub.ax/PXWeb/pxweb/sv/Statistik/Statistik\\_\\_FO\\_\\_F%C3%B6retagens%20ekonomi/F0013b.px/](https://pxweb.asub.ax/PXWeb/pxweb/sv/Statistik/Statistik__FO__F%C3%B6retagens%20ekonomi/F0013b.px/).
- [16] ÅSUB. Näringslivets löneutbetalningar (1 000 euro) efter år, månad och bransch, 2024. URL [https://pxweb.asub.ax/PXWeb/pxweb/sv/Statistik/Statistik\\_\\_FO\\_\\_F%C3%B6retagens%20ekonomi/F0014b.px/](https://pxweb.asub.ax/PXWeb/pxweb/sv/Statistik/Statistik__FO__F%C3%B6retagens%20ekonomi/F0014b.px/).
- [17] Gareth James, Daniela Witten, Trevor Hastie, Robert Tibshirani, and Jonathan Taylor. *An Introduction to Statistical Learning: with applications in Python*. Springer, May 2023. URL <https://www.statlearning.com>.
- [18] Marius-Constantin Popescu, Valentina E Balas, Liliana Perescu-Popescu, and Nikos Mastorakis. Multi-layer Perceptron and Neural Networks. *WSEAS Transactions on Circuits and Systems*, 8(7), 2009.
- [19] A D Dongare, R R Kharde, and Amit D Kachare. Introduction to Artificial Neural Network. *International Journal of Engineering and Innovative Technology*, 2(1), 2012.
- [20] Siddharth Sharma, Simone Sharma, and Anidhya Athaiya. ACTIVATION FUNCTIONS IN NEURAL NETWORKS. *International Journal of Engineering Applied Sciences and Technology*, 04(12):310–316, May 2020. ISSN 24552143. doi: 10.33564/IJEAST.2020.v04i12.054. URL [https://www.ijeast.com/papers/310-316\\_Tesma412\\_IJEAST.pdf](https://www.ijeast.com/papers/310-316_Tesma412_IJEAST.pdf).
- [21] Diederik P. Kingma and Jimmy Ba. Adam: A Method for Stochastic Optimization, January 2017. URL <http://arxiv.org/abs/1412.6980>. arXiv:1412.6980 [cs].
- [22] Aurélien Géron. *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow*. O’Reilly, 3rd edition, 2022. ISBN 978-1-09-812597-4.
- [23] Alex Kendall and Yarin Gal. What Uncertainties Do We Need in Bayesian Deep Learning for Computer Vision?, October 2017. URL <http://arxiv.org/abs/1703.04977>. arXiv:1703.04977 [cs].

- [24] Yarin Gal and Zoubin Ghahramani. Dropout as a Bayesian Approximation: Representing Model Uncertainty in Deep Learning, October 2016. URL <http://arxiv.org/abs/1506.02142>. arXiv:1506.02142 [cs, stat].
- [25] Mim Kemal Tekin. *Vehicle Path Prediction Using Recurrent Neural Networks*. Examensarbete, Linköping, Linköping, 2020.
- [26] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural Computation*, 9(8):1735–1780, November 1997. ISSN 0899-7667. doi: 10.1162/neco.1997.9.8.1735.
- [27] Kyunghyun Cho, Bart van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation, September 2014. URL <http://arxiv.org/abs/1406.1078>. arXiv:1406.1078 [cs, stat].
- [28] ÅSUB. Konjunkturläget våren 2018. (2018):62, 2018. ISSN 1455–1977. URL [https://www.asub.ax/sites/default/files/reports/konjunkturlaget\\_varen\\_2018.pdf](https://www.asub.ax/sites/default/files/reports/konjunkturlaget_varen_2018.pdf).
- [29] ÅSUB. Konjunkturläget våren 2019. (2019):56, 2019. ISSN 1455–1977. URL [https://www.asub.ax/sites/default/files/reports/konjunkturlaget\\_varen\\_2019.pdf](https://www.asub.ax/sites/default/files/reports/konjunkturlaget_varen_2019.pdf).
- [30] Johan Flink. Ekonomisk översikt hösten 2019. (2019):17, 2019. URL [https://www.asub.ax/sites/default/files/reports/ekonomisk\\_oversikt\\_hosten\\_2019\\_-\\_tillganglig.pdf](https://www.asub.ax/sites/default/files/reports/ekonomisk_oversikt_hosten_2019_-_tillganglig.pdf).
- [31] ÅSUB. Konjunkturläget våren 2023. (2023:1):70, 2023. ISSN 1455-1977. URL [https://www.asub.ax/sites/default/files/media/document/Konjunktur1%C3%A4get%20v%C3%A5ren%202023\\_1.pdf](https://www.asub.ax/sites/default/files/media/document/Konjunktur1%C3%A4get%20v%C3%A5ren%202023_1.pdf).
- [32] Lingxue Zhu and Nikolay Laptev. Deep and Confident Prediction for Time Series at Uber. In *2017 IEEE International Conference on Data Mining Workshops (ICDMW)*, pages 103–110, November 2017. doi: 10.1109/ICDMW.2017.19. URL <http://arxiv.org/abs/1709.01907>. arXiv:1709.01907 [stat].
- [33] Sabeen Ahmed, Ian E. Nielsen, Aakash Tripathi, Shamooun Siddiqui, Ghulam Rasool, and Ravi P. Ramachandran. Transformers in Time-series Analysis: A Tutorial. *Circuits, Systems, and Signal Processing*, 42(12):7433–7466, December 2023. ISSN 0278-081X, 1531-5878. doi: 10.1007/s00034-023-02454-8. URL <http://arxiv.org/abs/2205.01138>. arXiv:2205.01138 [cs].