Neural Networks in Reserving

Jan Kościałkowski, Kamil Kuźmicki, Anna Przybyła

Polish Society of Actuaries

December 2020

POLSKIE
STOWARZYSZENIE
AKTUARIUSZY

Introduction

We covered two papers, one related to claims data simulation and the other discussing a simple use of neural networks to predict future values in a loss triangle.

- Andrea Gabrielli, Mario V. Wüthrich, *Individual Claims History Simulation Machine*, Risks, 2018.
  This paper proposes a neural-network-based simulation machine for generating claims data. Multiple models are trained to be able to generate data with distribution consistent with that of the real-world training data.
- Mario V. Wüthrich, *Neural Networks Applied to Chain-Ladder Reserving*, European Actuarial Journal, 2018.
  A neural networks are used to predict future values in a loss triangle. Data is aggregated per distinct value of explanatory variable, a simple model is used for triangles with 0 claims and a separate neural network is trained for each development period.

Input Data

All models were trained on a real dataset with $\sim$ 10M individual claims histories. The following variables were avaialble:

- `ClNr` - a distinct claims identifier
- `LoB` - the line of business, $\in \{1,\ldots,4\}$
- `cc` - denotes the labor sector of the injured, $\in \{1,\ldots,53\}$
- `AY` - accident year, $\in \{1994,\ldots,2005\}$
- `AQ` - accident quarter, $\in \{1,\ldots,4\}$
- `age` - the age of the injured, in 5 years age buckets, $\in \{15,20,\ldots,70\}$
- `inj_part` - denotes the injured body part, $\in \{1,\ldots,99\}$
- `RY` - the reporting year, $\in \{1994,\ldots,2016\}$

Moreover, the data contains claim cash flows in subsequent years and a settlement indicator.

Simulation Machine Design

Multiple neural networks are trained to be able to recover the claims history, each subsequent network basing on the predictions of the previous one. It was done in 8 general steps:

1. reporting delay T simulation
2. payment indicator Z simulation
3. number of payments K simulation
4. total claim size Y simulation
5. number of recovery payments K^- simulation
6. recovery size Y^- simulation
7. cash flow $(C_i^{(j)})_{0 \leq j \leq 11}$ simulation
8. claim status $(I_i^{(j)})_{0 \leq j \leq 11}$ simulation

Simulation Summary

- The main paper used a dataset with 5M rows, so we wanted to use a similar one as well.
- Initially, a laptop with 24 GB RAM was used, but R ran out of memory.
- Finally, the simulation took 10 minutes on an `m5.12xlarge` AWS machine (48 cores, 192 GB RAM).
- Datasets with 500k rows were also generated, because the one with 5M rows caused memory problems.

Model Assumptions

Mack's Chain Ladder method

$$C_{i,j} \approx f_{j-1} * C_{i,j-1}$$

$C_{i,j}$ denotes the cumulative claims payments for claims with accident year i which are done within the first j development years.
f_j are CL factors (not depend on accident year).

Feature space $\mathcal{X}$ with features $x \in \mathcal{X}$ describing the characteristics of individual claims:

- the line of business $LoB \in \{1, \dots, 4\}$;
- the claims code $cc \in \{1, \dots, 53\}$ denoting the labor sector the injured is working in;
- the accident quarter $AQ \in \{1, \dots, 4\}$;
- the age of the injured $age \in \{15, \dots, 70\}$;
- the injured body part $inj\_part \in \{10, \dots, 99\}$.

Model Assumptions

Model assumptions:

- 5-dimensional feature space $\mathcal{X}$ collecting the features $x = (x_1, \ldots, x_5)' = (LoB, cc, AQ, age, inj\_part)'$.
- Cumulative claims $C_{i,j}(x)$ are non-negative.
- Cumulative claims $C_{i,j}(x)$ of different accident years $1 \leq i \leq I$ or different features $x \in \mathcal{X}$ are independent.
- There exist parameters $f_0(x), \ldots, f_{J-1}(x) > 0$ such that for all $1 \leq i \leq I, 1 \leq j \leq J$ and $x \in \mathcal{X}$

$$\mathbb{E}[C_{i,j}(x)|\mathcal{F}_{i+j-1}] = f_{j-1}(x)C_{i,j-1}(x) + \mathbb{E}[C_{i,j}(x)|\mathcal{F}_{i+j-1}]1_{\{C_{i,j-1}(x)=0\}}$$

- There exist parameters $\sigma_0^2(x), \ldots, \sigma_{J-1}^2(x) > 0$ such that for all $1 \leq i \leq I, 1 \leq j \leq J$ and $x \in \mathcal{X}$

$$Var(C_{i,j}(x)|\mathcal{F}_{i+j-1}) = \sigma_{j-1}^2(x)C_{i,j-1}(x) + Var[C_{i,j}(x)|\mathcal{F}_{i+j-1}]1_{\{C_{i,j-1}(x)=0\}}$$

Model Assumptions

Then for $C_{i,j-1}(x) > 0$:

$$\mathbb{E}[C_{i,j}(x)|\mathcal{F}_{i+j-1}] = f_{j-1}(x)C_{i,j-1}(x).$$

For $C_{i,j-1}(x) = 0$

$$\mathbb{E}[C_{i,j}|\mathcal{F}_{i+j-1}] = \mathbb{E}[\sum_{x^+\in\mathcal{X}^+} C_{i,j}(x^+)|\mathcal{F}_{i+j-1}] = \sum_{x^+\in\mathcal{X}^+} f_{j-1}(x^+)C_{i,j-1}(x^+).$$

Loss Function

The initially considered loss function is taken to be the weighted square loss.

$$\mathcal{L}_j = \sum_{i=1}^{I-j} \sum_{\mathbf{x}: C_{i,j-\mathbf{1}}(\mathbf{x})>0} \frac{\left(C_{i,j}(\mathbf{x}) - f_{j-1}(\mathbf{x}) C_{i,j-1}(\mathbf{x})\right)^2}{\sigma_{j-1}^2 C_{i,j-1}(\mathbf{x})}$$

$$= \frac{1}{\sigma_{j-1}^2} \sum_{i=1}^{I-j} \sum_{\mathbf{x}: C_{i,j-\mathbf{1}}(\mathbf{x})>0} C_{i,j-1}(\mathbf{x}) \left(\frac{C_{i,j}(\mathbf{x})}{C_{i,j-1}(\mathbf{x})} - f_{j-1}(\mathbf{x}) \right)^2$$

The training can be simplified by a slight modification.

$$\mathcal{L}_j^0 = \sigma_{j-1}^2 \mathcal{L}_j = \sum_{i=1}^{I-j} \sum_{\mathbf{x}: C_{i,j-\mathbf{1}}(\mathbf{x})>0} \left(\frac{C_{i,j}(\mathbf{x})}{\sqrt{C_{i,j-1}(\mathbf{x})}} - f_{j-1}(\mathbf{x}) \sqrt{C_{i,j-1}(\mathbf{x})} \right)^2$$

Training Procedure

Categorical variables (*LoB*, *cc* and *inj_part*) were one-hot-encoded and numerical (*AQ*, *age*) were min-max-scaled. One neural network was trained per development year.

Neural network settings:

- One hidden layer with $q = 20$ neurons
- Batch size 10000
- 100 epochs
- Optimized with RMSProp with 0.001 learning rate

Zero claims features - Problem

Let us fix accident year i.

We now turn to the case $C_{i,I-i}(x) = 0$ called zero claims features. It corresponds to the situation that for a given features' combination x there have been no payments observed so far. But we can surely expect that there will be some in the next years.

Our goal is to predict future total amount of payments out of all such claims that so far had no payments.

The idea here is to predict the amount of first payments out of these claims in the next year (i.e. for $I - i + 1$-th development year) and then, using Chain-Ladder-like reasoning, carry it forward.

Zero claims features - Solution

Let $C^*_{k,j}$, for $j \geq I - i$, be the cumulative sum of payments from claims of accident year k, which for $j = I - i$ had zero payments.

The assumption of the model is that:

$$\frac{C^*_{i,I-i+1}}{C_{i,I-i}} := \frac{\sum_{k=1}^{i-1} C^*_{k,I-i+1}}{\sum_{k=1}^{i-1} C_{k,I-i}}$$

And then (for $J > j > I - i$):

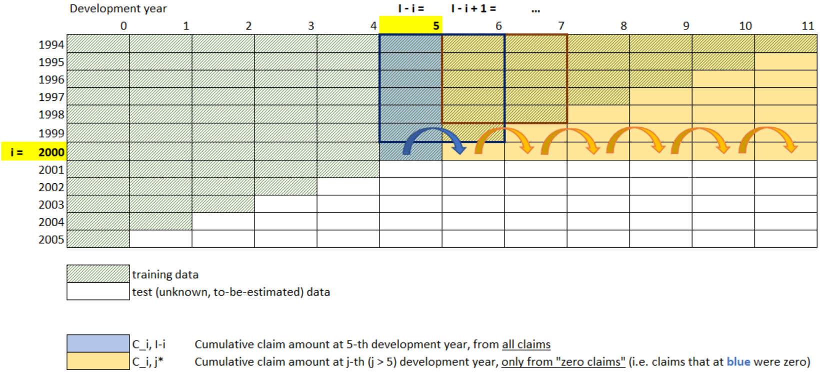
$$\frac{C^*_{i,j+1}}{C^*_{i,j}} := \frac{\sum_{k=1}^{I-j-1} C^*_{k,j+1}}{\sum_{k=1}^{I-j-1} C^*_{k,j}}$$

Note that right hand-side of the equations and $C_{i,I-i}$ is always fully known. Based on this, we can calculate $C^*_{i,J}$ iteratively.

Zero claims features - Example for $i = 2000$

Let us study an example with $i = 2000$ and $I = 2005$.

We build an artifical "triangle", where each cell contains the cumulative sum of claim amounts from: a) all claims for blue cells and b) only "zero claims" for yellow cells.



Results - 500k

	line of business **LoB 1**		
	true reserves	**NN reserves**	**CL reserves**
1994	0	0	0
1995	55	138	76
1996	253	278	185
1997	221	387	276
1998	251	547	400
1999	891	818	656
2000	766	903	817
2001	1 122	1 241	1 202
2002	1 608	1 673	1 644
2003	2 572	2 545	2 634
2004	3 860	4 014	3 953
2005	8 387	8 097	8 035
total	**19 987**	**20 640**	**19 879**

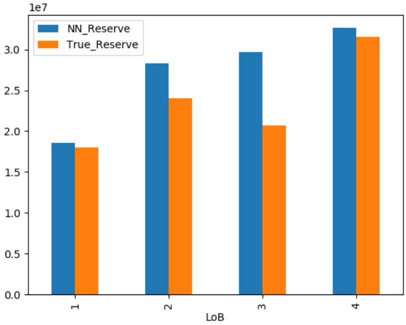
	line of business **LoB 2**		
	true reserves	**NN reserves**	**CL reserves**
1994	0	0	0
1995	2	133	-23
1996	86	360	43
1997	449	588	170
1998	316	755	307
1999	360	967	511
2000	195	809	702
2001	774	1 226	1 075
2002	2 283	2 423	1 915
2003	3 022	3 255	3 231
2004	6 105	6 573	6 320
2005	15 501	17 593	16 087
total	**29 094**	**34 682**	**30 338**

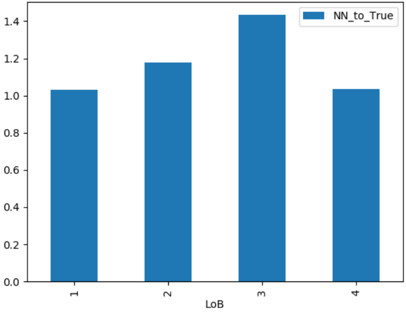
	line of business **LoB 3**		
	true reserves	**NN reserves**	**CL reserves**
1994	0	0	0
1995	47	173	66
1996	189	301	156
1997	289	679	287
1998	797	1 162	470
1999	122	1 161	603
2000	1 212	1 367	1 017
2001	1 363	2 081	1 510
2002	1 684	2 731	2 042
2003	3 170	3 822	3 475
2004	4 645	6 394	5 916
2005	12 308	16 187	16 038
total	**25 825**	**36 057**	**31 579**

	line of business **LoB 4**		
	true reserves	**NN reserves**	**CL reserves**
1994	0	0	0
1995	126	181	93
1996	297	420	239
1997	857	818	471
1998	969	1 106	721
1999	966	1 418	1 099
2000	1 409	1 690	1 413
2001	1 786	2 306	2 155
2002	3 247	3 370	3 214
2003	5 693	4 793	5 044
2004	7 822	7 104	7 280
2005	11 225	12 544	12 905
total	**34 399**	**35 749**	**34 634**

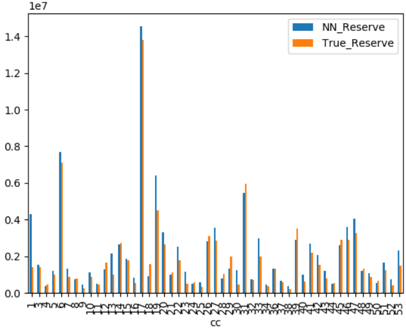
Figures in (000).

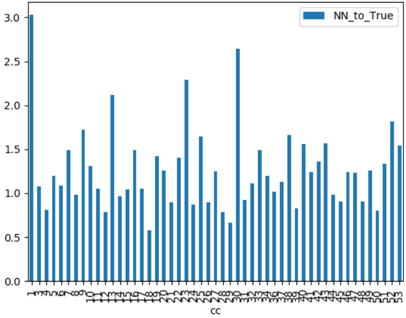
Results - 500k - LoB



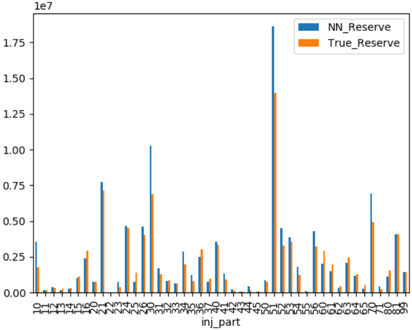


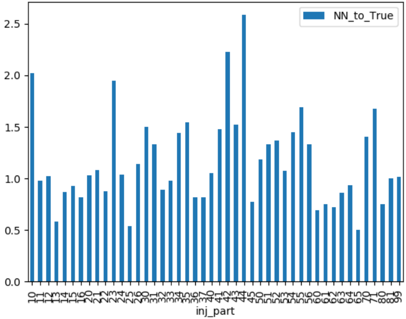
Results - 500k - cc



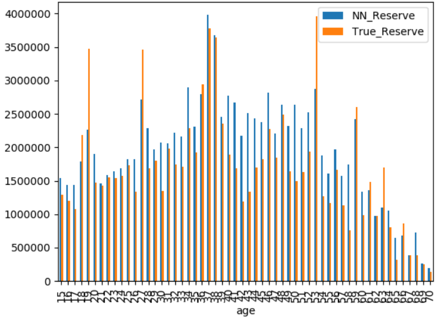


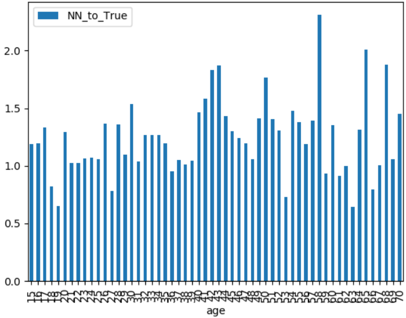
Results - 500k - inj_part





Results - 500k - age





Weights initialization

As the results for 500k seemed not very stable, we tried to test some weights initialization.

Our approach was to set initial weights in the neural networks in such a way that, before training, the network should output averaged, Chain-Ladder-like results.

It was implemented by setting in the output layer:

- the same Chain-Ladder-like, initial bias for each data point,
- other initial weights equal to zero (to ignore input from previous, randomly initialized layers).

The results suggest that for smaller datasets this approach might improve stability.

Results - 500k - Weights initialization

	line of business **LoB 1**		
	true reserves	**NN reserves**	**CL reserves**
1994	0	0	0
1995	55	90	76
1996	253	202	185
1997	221	265	276
1998	251	396	400
1999	891	696	656
2000	766	869	817
2001	1 122	1 217	1 202
2002	1 608	1 603	1 644
2003	2 572	2 555	2 634
2004	3 860	4 018	3 953
2005	8 387	8 036	8 035
total	**19 987**	**19 947**	**19 879**

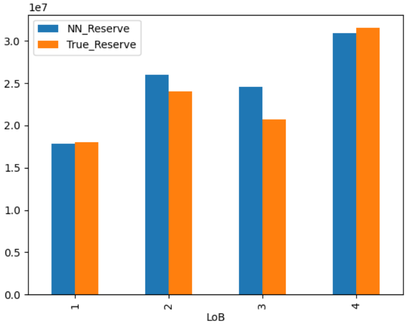
	line of business **LoB 2**		
	true reserves	**NN reserves**	**CL reserves**
1994	0	0	0
1995	2	15	-23
1996	86	69	43
1997	449	223	170
1998	316	305	307
1999	360	765	511
2000	195	930	702
2001	774	1 341	1 075
2002	2 283	2 200	1 915
2003	3 022	3 107	3 231
2004	6 105	6 244	6 320
2005	15 501	17 161	16 087
total	**29 094**	**32 359**	**30 338**

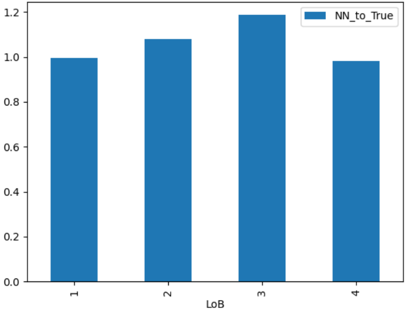
	line of business **LoB 3**		
	true reserves	**NN reserves**	**CL reserves**
1994	0	0	0
1995	47	53	66
1996	189	115	156
1997	289	265	287
1998	797	587	470
1999	122	849	603
2000	1 212	1 235	1 017
2001	1 363	1 667	1 510
2002	1 684	2 264	2 042
2003	3 170	3 320	3 475
2004	4 645	5 694	5 916
2005	12 308	14 895	16 038
total	**25 825**	**30 945**	**31 579**

	line of business **LoB 4**		
	true reserves	**NN reserves**	**CL reserves**
1994	0	0	0
1995	126	104	93
1996	297	238	239
1997	857	471	471
1998	969	817	721
1999	966	1 125	1 099
2000	1 409	1 463	1 413
2001	1 786	2 165	2 155
2002	3 247	3 231	3 214
2003	5 693	4 842	5 044
2004	7 822	7 224	7 280
2005	11 225	12 378	12 905
total	**34 399**	**34 057**	**34 634**

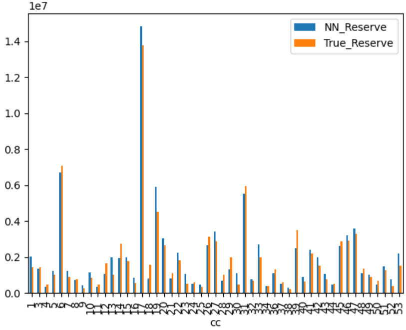
Figures in (000).

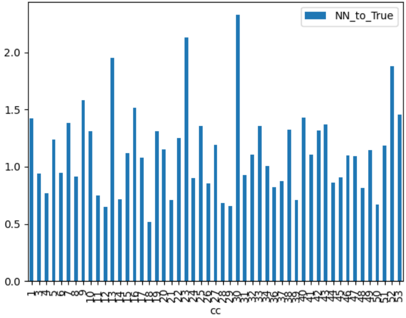
Results - 500k - Weights initialization - LoB



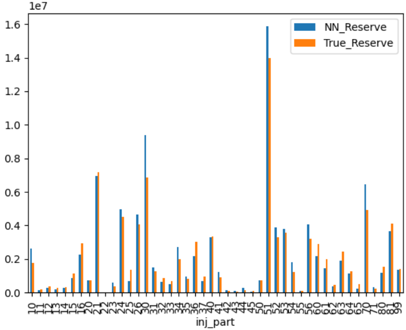


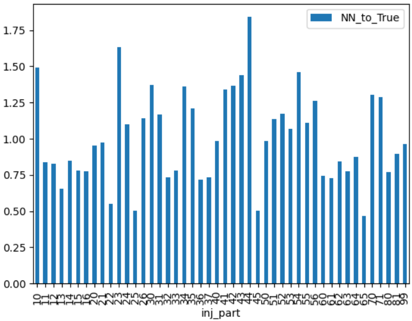
Results - 500k - Weights initialization - cc



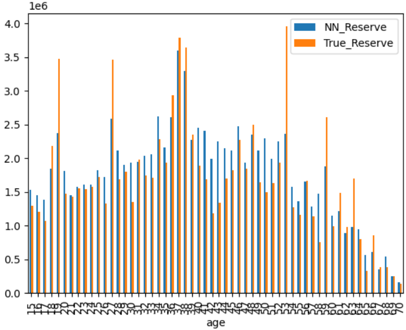


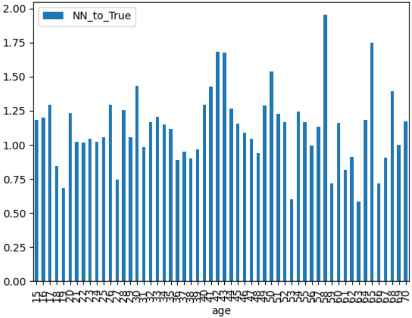
Results - 500k - Weights initialization - inj_part





Results - 500k - Weights initialization - age





Results - 5M

	line of business **LoB 1**		
	true reserves	**NN reserves**	**CL reserves**
1994	0	0	0
1995	1 048	900	1 039
1996	2 038	1 888	2 276
1997	3 807	2 875	3 818
1998	4 951	3 880	5 574
1999	8 190	6 703	7 938
2000	12 167	10 618	11 335
2001	16 918	15 884	16 130
2002	21 894	20 627	22 132
2003	35 075	32 176	33 284
2004	55 008	49 020	51 290
2005	110 414	100 385	102 508
total	**271 511**	**244 956**	**257 324**

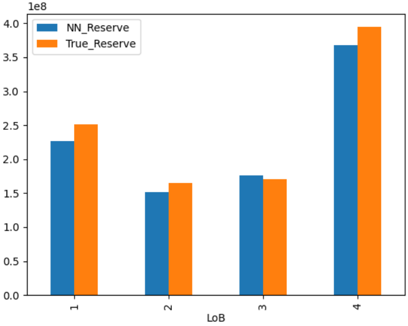
	line of business **LoB 2**		
	true reserves	**NN reserves**	**CL reserves**
1994	0	0	0
1995	-16	-558	-218
1996	472	-373	293
1997	1 210	-141	982
1998	1 797	-90	2 009
1999	4 483	1 894	3 098
2000	4 974	4 111	4 565
2001	6 940	7 114	6 921
2002	12 675	11 354	11 584
2003	20 808	18 029	19 418
2004	44 554	40 950	43 048
2005	100 068	99 153	99 289
total	**197 965**	**181 441**	**190 989**

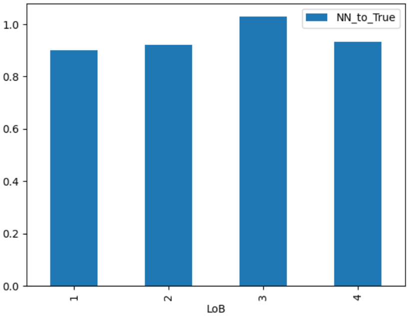
	line of business **LoB 3**		
	true reserves	**NN reserves**	**CL reserves**
1994	0	0	0
1995	599	67	302
1996	1 507	675	1 211
1997	2 847	1 201	2 258
1998	3 029	1 527	3 569
1999	4 084	3 826	4 926
2000	5 693	6 166	6 737
2001	9 536	10 601	10 666
2002	13 093	13 904	14 678
2003	21 247	23 985	24 617
2004	46 284	44 650	46 741
2005	110 932	106 353	107 261
total	**218 850**	**212 956**	**222 966**

	line of business **LoB 4**		
	true reserves	**NN reserves**	**CL reserves**
1994	0	0	0
1995	1 931	1 580	1 857
1996	4 277	3 516	4 110
1997	5 753	5 186	6 445
1998	9 466	7 805	9 811
1999	14 273	12 566	14 288
2000	20 387	18 720	19 906
2001	26 931	26 570	27 222
2002	39 586	37 635	39 095
2003	54 792	52 220	55 042
2004	81 885	78 658	82 795
2005	162 940	152 757	155 611
total	**422 220**	**397 212**	**416 183**

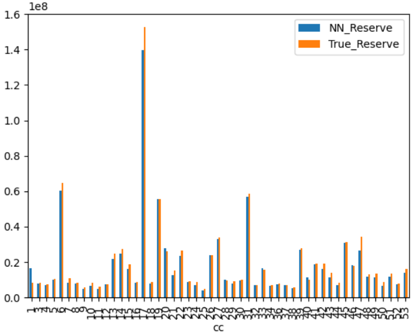
Figures in (000).

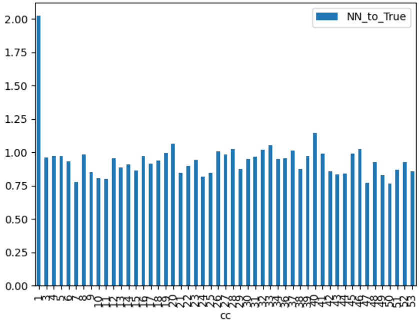
Results - 5M - LoB



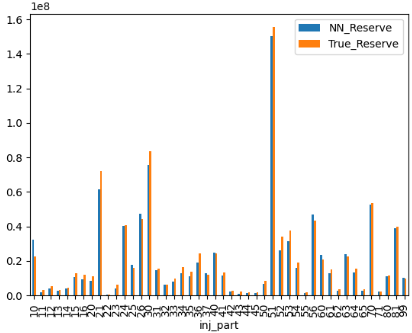


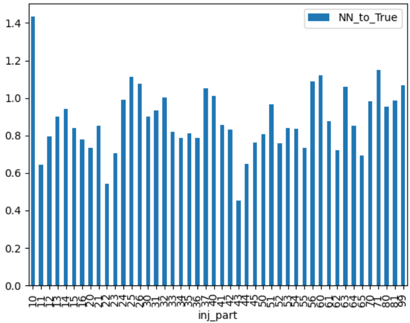
Results - 5M - cc



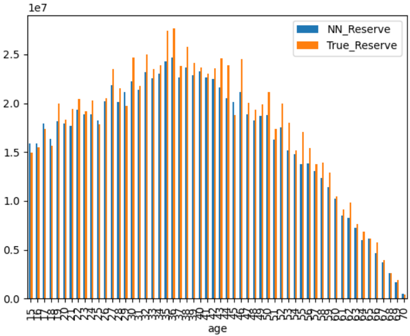


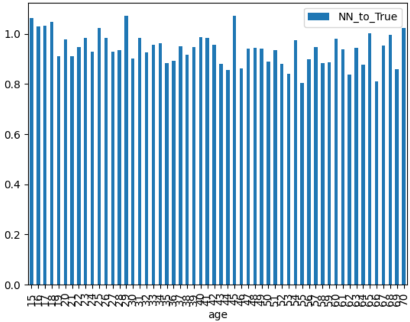
Results - 5M - inj_part





Results - 5M - age





Discrepancies

There were multiple differences in the training process between the paper and our experience:

- The training times were substantially longer ($\sim$4 mins on a GPU vs $\sim$ 2 mins on CPU).
- For lower j, the models seemed untrained after 100 epochs, MSEs were much higher than reported in the paper (3000 vs 2 for $j = 1$).

Next steps

- Aggregate reserves
  - Change loss function - y's concentrated in 1, normal distribution probably inadequate
  - Consider some optimization of hyper-parameters, e.g. number of epochs, number of neurons etc.
  - Apply XAI to the neural networks from the discussed paper
  - DeepTriangle - RNN-based neural network trained on loss triangles
- Individual reserves