Quantee

quantee.ai

The use of denoising autoencoders for categorical and continuous variables

Anna Kozak[1]
joint work with Łukasz Delong[1,2]

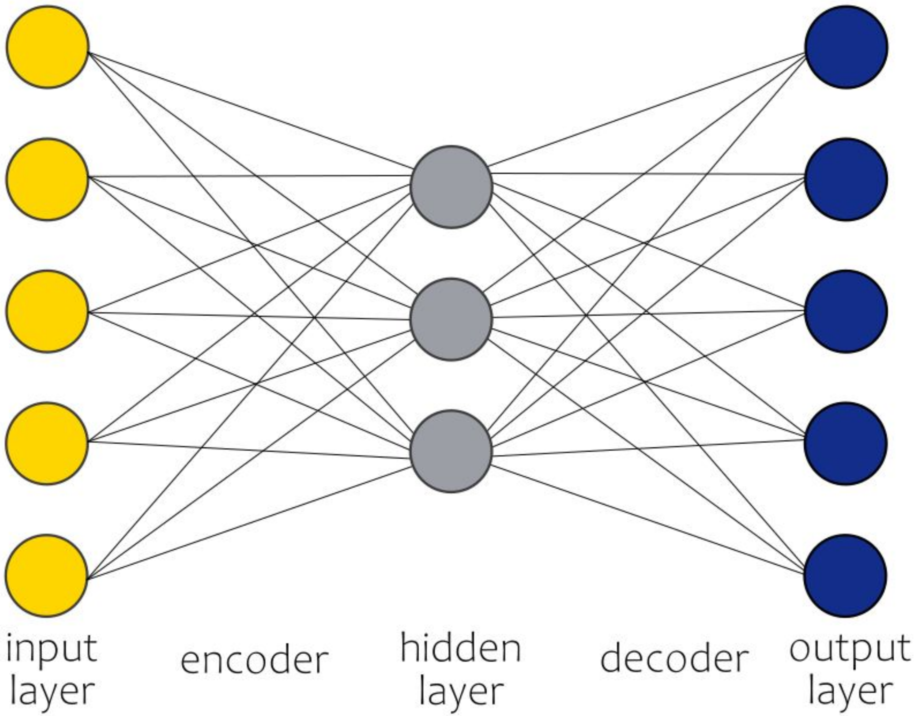
[1] Quantee, R&D Department

[2] SGH Warsaw School of Economics, Institute of Econometrics

June 17, 2021

Test autoencoders on categorical variables, until now Entity Embedding has been used in the actuarial field.

Autoencoders



Benefits of using autoencoders

Erhan, Dumitru and Manzagol, Pierre-Antoine and Bengio, Y. and Bengio, S. and Vincent, Pascal . (2009). *The Difficulty of Training Deep Architectures and the Effect of Unsupervised Pre-Training.* Twelfth International Conference on Artificial Intelligence and Statistics

Vincent, Pascal and Larochelle, Hugo and Lajoie, Isabelle and Bengio, Y. and Manzagol, Pierre-Antoine. (2008). *Extracting and composing robust features with denoising autoencoders.* In ICML 2008: Proceedings of the Twenty-fifth International Conference on Machine Learning

Vincent, Pascal and Larochelle, Hugo and Lajoie, Isabelle and Bengio, Y. and Manzagol, Pierre-Antoine. (2010). *Stacked Denoising Autoencoders: Learning Useful Representations in a Deep Network with a Local Denoising Criterion.*, Journal of Machine Learning Research

Autoencoders for categorical variables

- autoencoder softmax all
- autoencoder softmax per variable

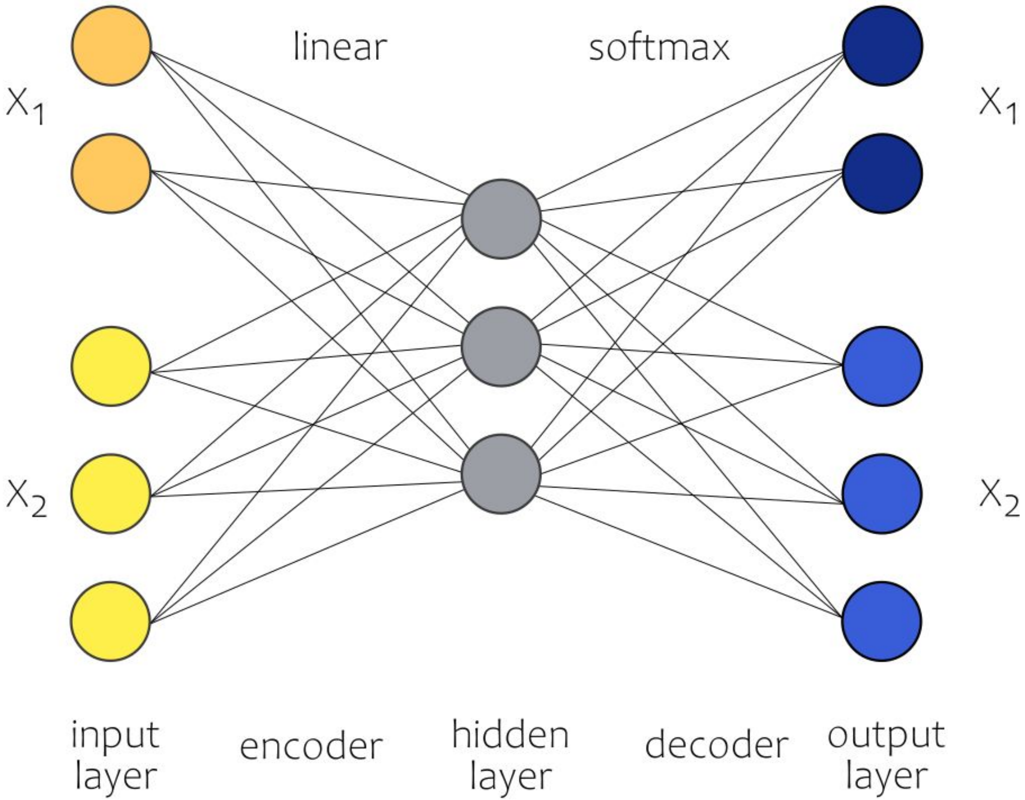
Categorical variables should be one-hot encoded before constructing the autoencoder.

We noise all observations and one, two or three randomly selected columns per observation.

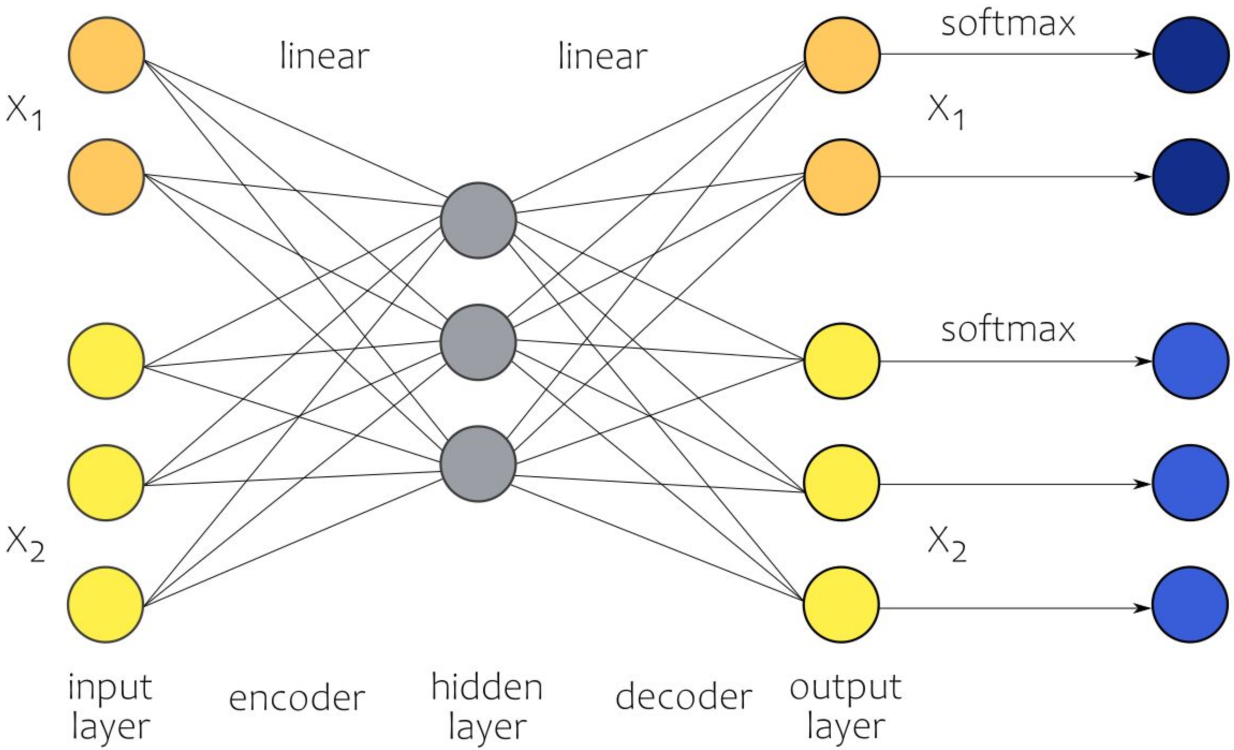
Each autoencoders we consider with noise. Types of noise:

- zeros 00100000 ⟶ 00000000
- sample 00100000 ⟶ 00000010

Autoencoder softmax all



Autoencoder softmax per variable



Autoencoders for numerical variables

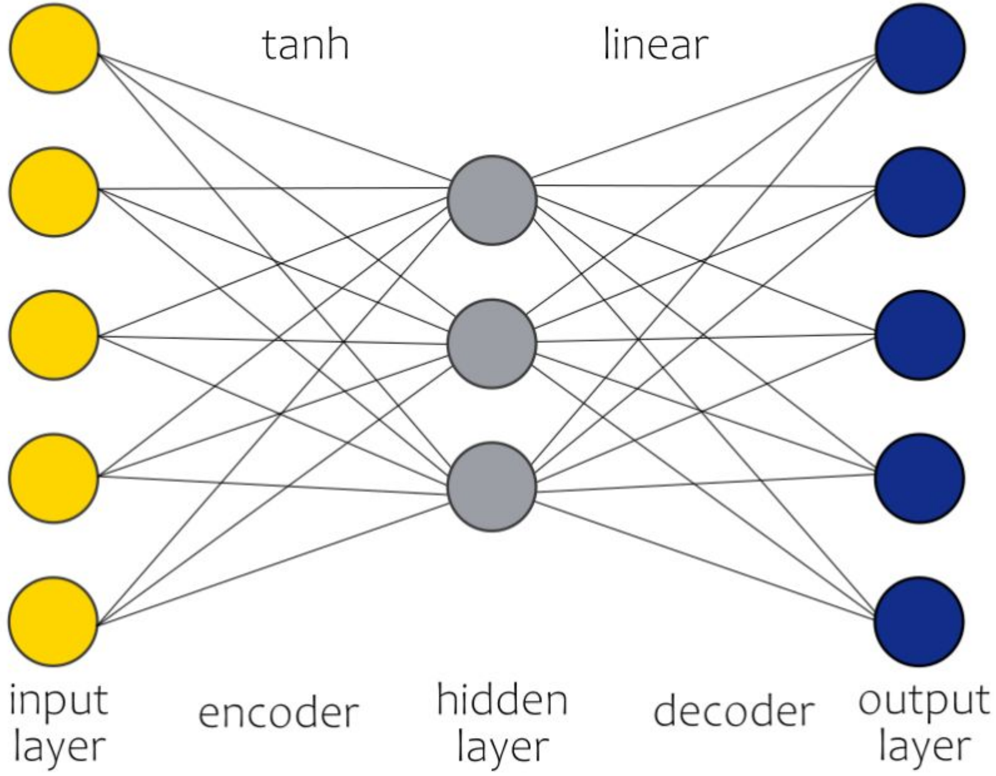
- autoencoder MSE

We noise all observations and one, two or three randomly selected columns per observation.

Each autoencoders we consider with noise. Types of noise:

- const - change value to zero
- sigma - change value to random value from normal distribution with expected value equal to the value from dataset and variance equal sigma square

Autoencoder for numerical variables



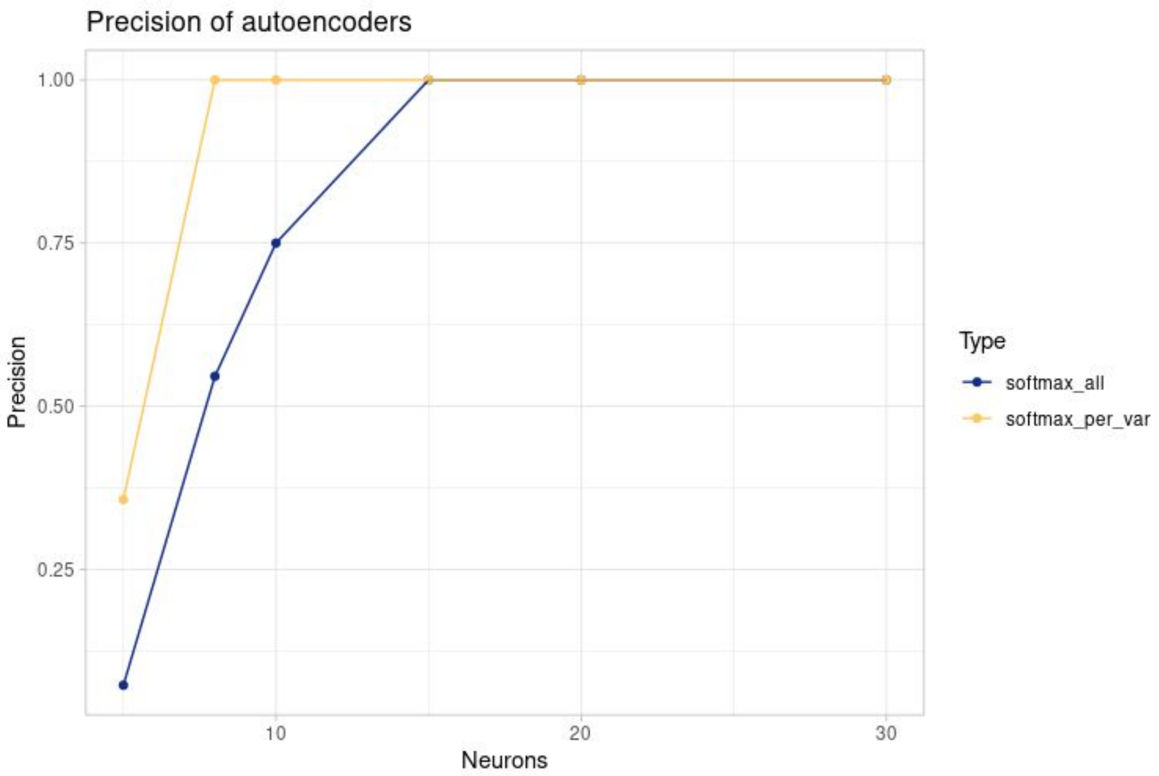
Dataset and experiments

Dataset: freMTPL2freq, select 100k observations by stratification
Categorical variable: Area, VehPower, VehAge, DrivAge, VehBrand, Region
Binary variable: VehGas
Numerical variable: BonusMalus, Density (log), Min-Max scaler

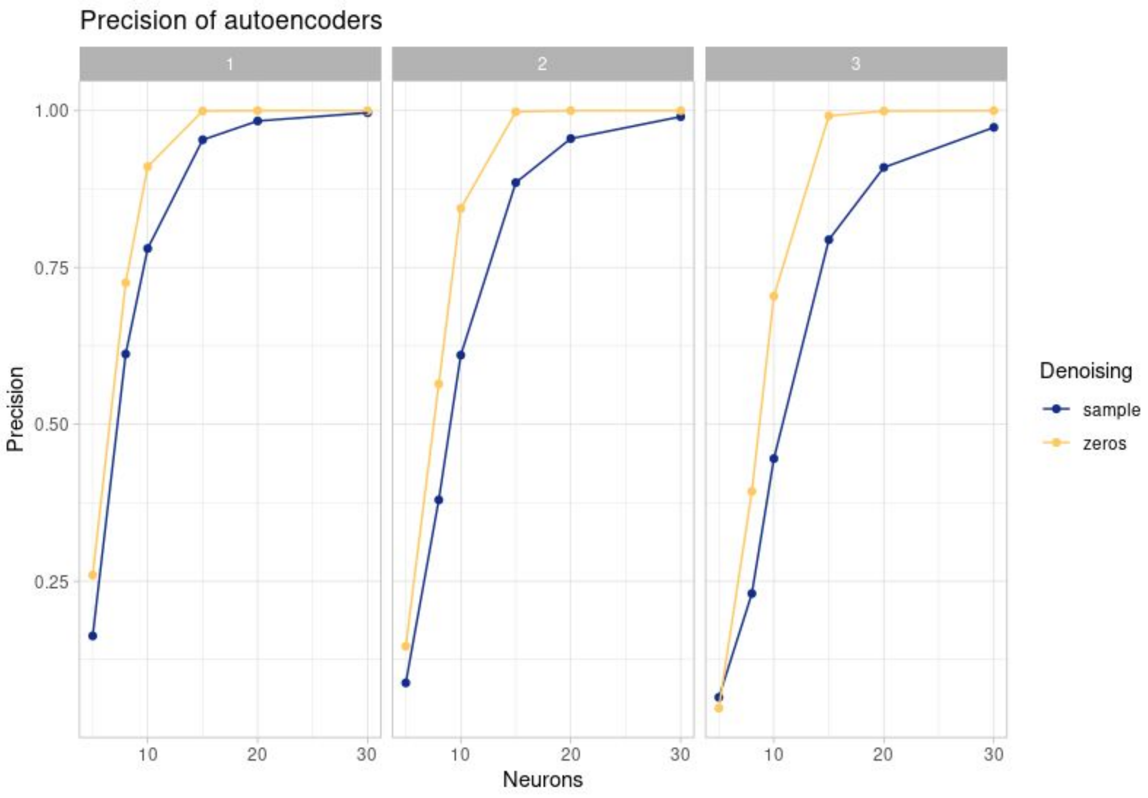
Experiments with autoencoder for categorical variables:

Parameter	Value
neurons	5, 8, 10, 15, 20, 30
epochs	500
batch size	1000
learning rate	0.001
min delta	0
patience	15
noise	zeros, sample

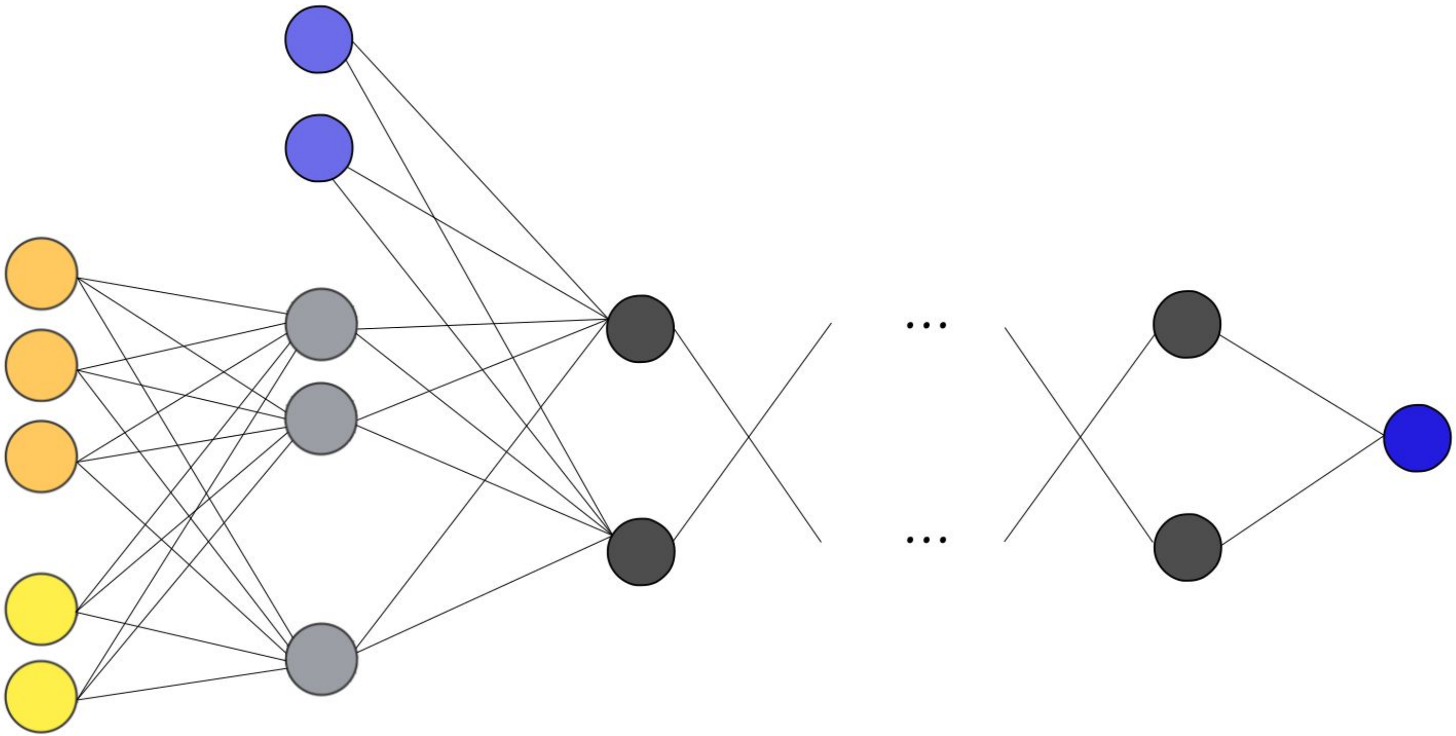
Results for autoencoders



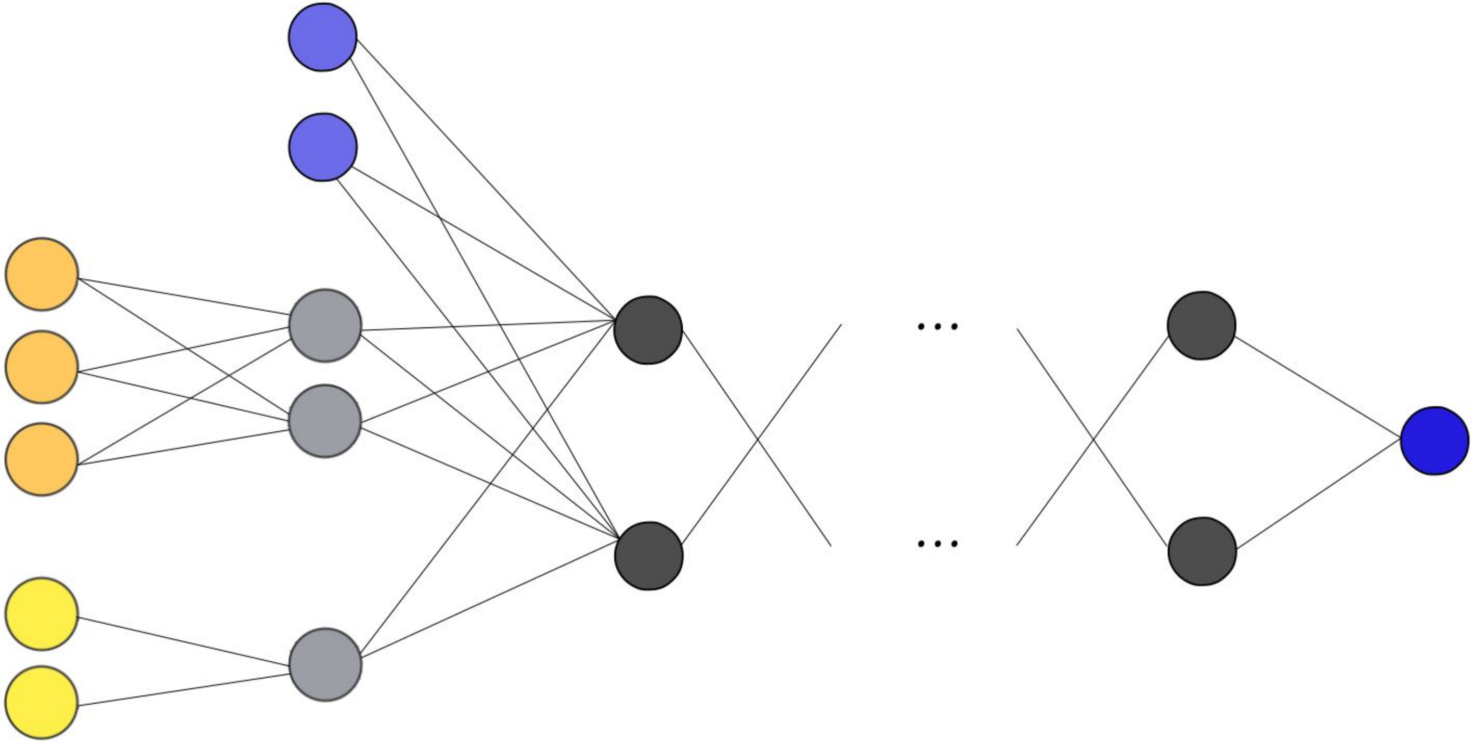
Results for autoencoder softmax per variable with denoising



Neural Networks with autoencoders



Neural Networks with Entity Embedding



Experiments

	With weight from autoencoder	
1 AE	epochs	15, 50, 100, 200, 300
	learning rate	5 * 10 ^ (-5), 5 * 10 ^ (-4), 5 * 10 ^ (-3)
	denoising	sample, zeros
	cols	1, 2, 3
2 AE	denoising	const, sigma
	sigma/feat noise	0, 0.1, 0.25, 0.5/ 0.1, 0.3, 0.5
	epochs	15, 50, 100, 200, 300
	learning rate	5 * 10 ^ (-5), 5 * 10 ^ (-4), 5 * 10 ^ (-3)
Neural Network	number layer	1, 3
	neurons layer	[20], [30], [50],[30, 30, 30], [30, 20, 10], [20, 15, 10]
	learning rate	10 ^(-4), 10 ^ (-3), 10 ^ (-2)

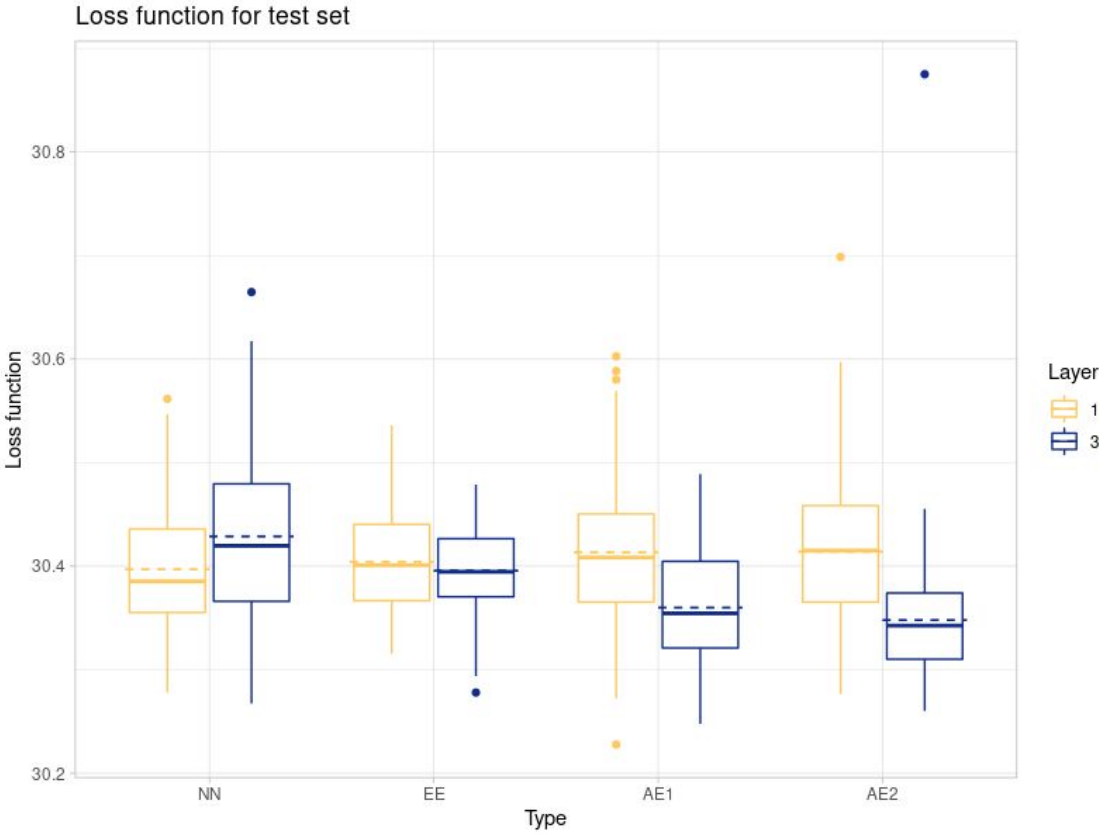
Without weight from autoencoder	
number layer	1, 3
neurons layer	[20], [30], [50],[30, 30, 30], [30, 20, 10], [20, 15, 10]
learning rate	10 ^ (-4), 10 ^ (-3), 10 ^ (-2)

Entity Embedding	
number layer	1, 3
neurons layer	[47], [57], [77], [25, 20, 11], [35, 24, 10], [33, 32, 32]
learning rate	10 ^ (-4), 10 ^ (-3), 10 ^ (-2)

Models chosen with 5 fold cv

neurons number	neurons_layer	number layer	learning rate AE	learinig rate	epochs	type	test
8	[20]	1	0.00005	0.01	100	2 AE	31,088
8	[50, 35, 20]	3	0.005	0.0001	15	2 AE	31,043
8	[20, 15, 10]	3	0.005	0.0001	200	2 AE	31,061
8	[50]	1	0.005	0.01	50	2 AE	31,100
8	[20]	1	0.00005	0.0001	15	1 AE	31,081
8	[20]	1	0.005	0.01	200	1 AE	31,081
8	[50]	1	0.0005	0.01	100	1 AE	31,083
8	[20, 15, 10]	3	0.0005	0.0001	100	1 AE	31,069
8	[50, 35, 20]	3	0.005	0.0001	15	1 AE	31,056
8	[50, 35, 20]	3	0.005	0.001	15	1 AE	31,061
8	[25, 20, 11]	3		0.001		EE	31,104
8	[50, 35, 20]	3		0.0001		NN	31,127
8	[30]	1		0.001		NN	31,097
8	[57]	1		0.01		EE	31,124

Results on test set - loss functions



Results on test set - loss functions

One layer

	EE	NN	AE1	AE2
1st Q	30,370	30,360	30,370	30,370
mean	**30,400**	**30,400**	**30,410**	**30,410**
3rd Q	30,440	30,440	30,450	30,460
sd	0,052	0,057	0,070	0,072

Three layer

	EE	NN	AE1	AE2
1st Q	30,370	30,370	30,320	30,310
mean	**30,400**	**30,430**	**30,360**	**30,350**
3rd Q	30,430	30,480	30,400	30,370
sd	0,040	0,083	0,053	0,068

Results on test set - claim intensities

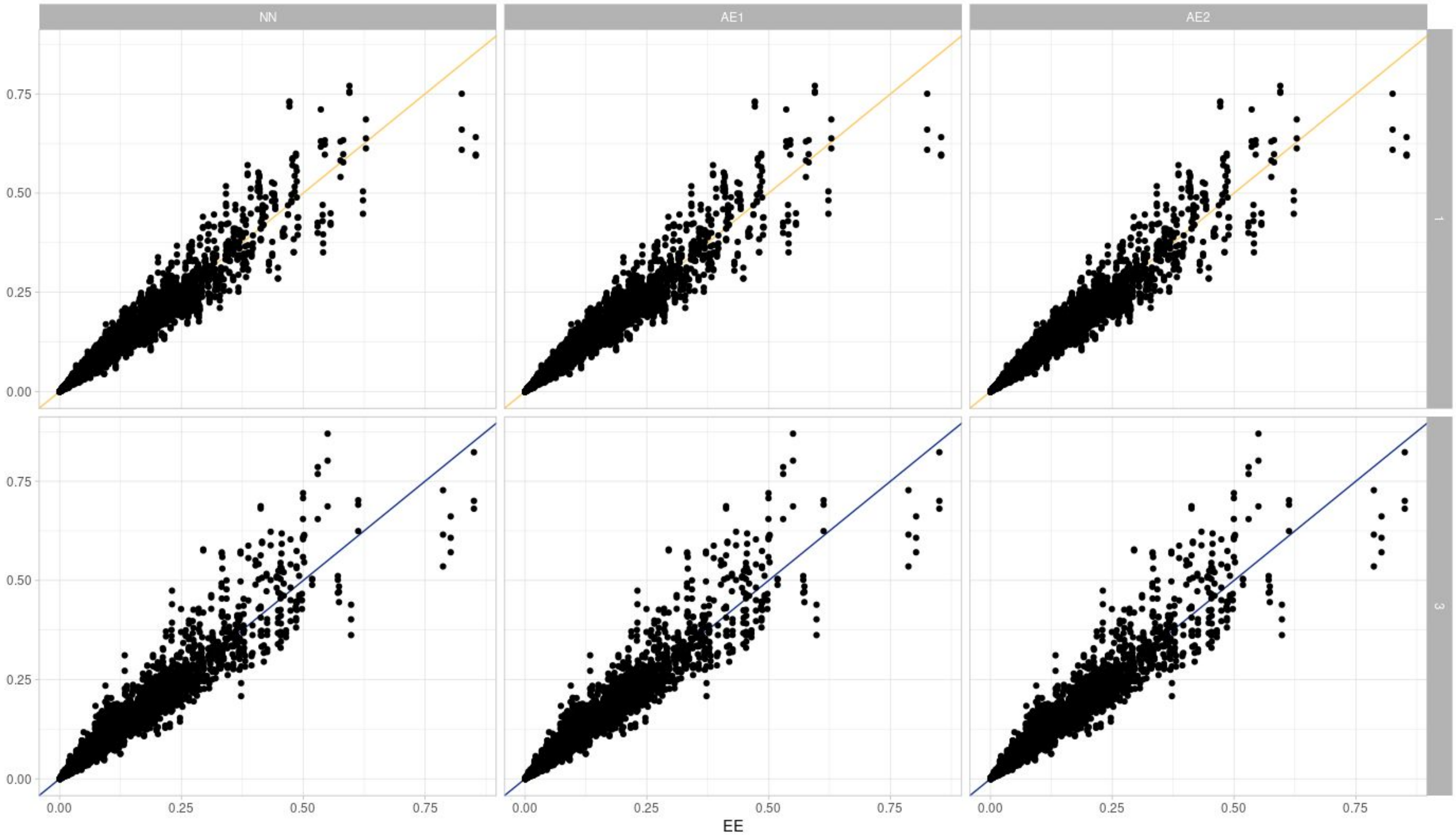
One layer

	NN	EE	AE1	AE2
1st Q	4,996	4,923	4,948	4,907
mean	**5,071**	**5,044**	**5,086**	**5,063**
3rd Q	5,157	5,17	5,236	5,178
sd	0,102	0,170	0,189	0,224
y mean	**5,105**	**5,105**	**5,105**	**5,105**

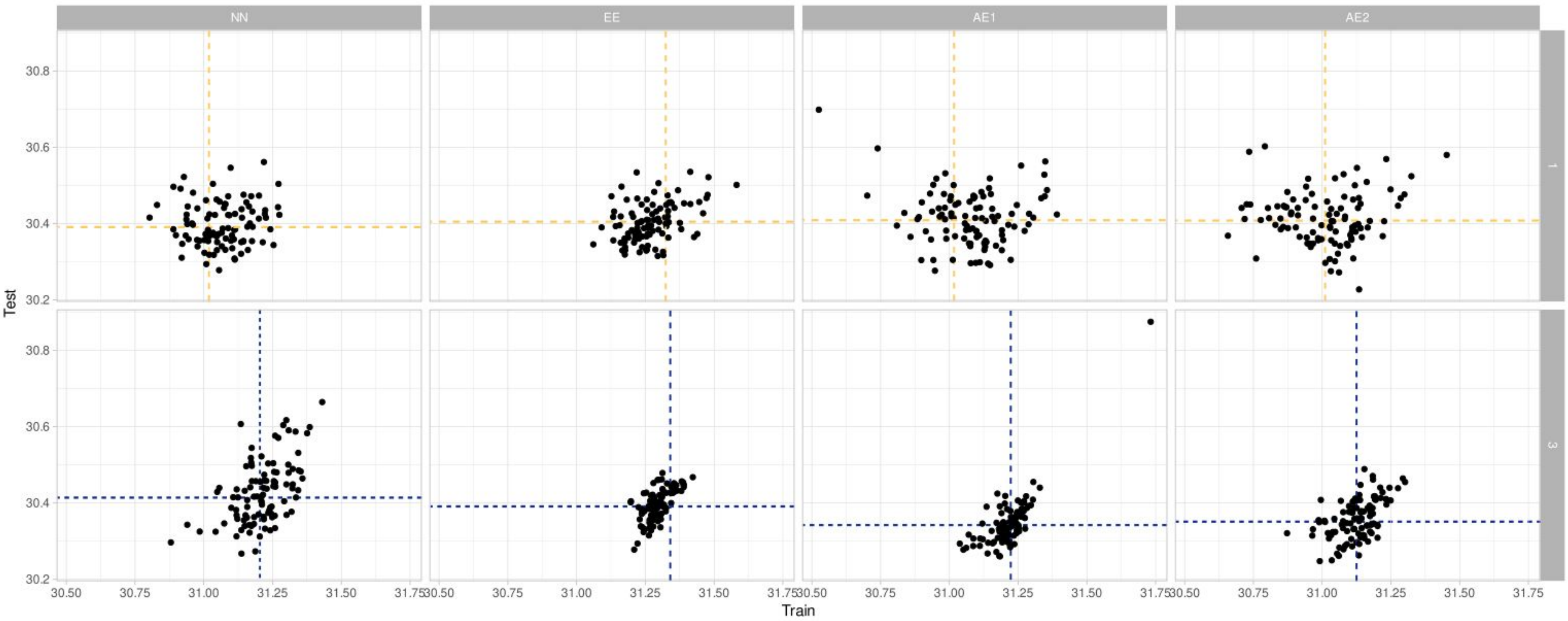
Three layer

	NN	EE	AE1	AE2
1st Q	5,099	4,981	5,141	5,234
mean	**5,149**	**5,054**	**5,183**	**5,262**
3rd Q	5,207	5,142	5,221	5,288
sd	0,068	0,114	0,060	0,039
y mean	**5,105**	**5,105**	**5,105**	**5,105**

Results on test set - claim intensities



Results on train and test sets - loss functions



What improvement comes with adding autoencoder?

AE	Loss function	Difference
0	30,408	
1	30,380	0,028
2	30,348	0,032

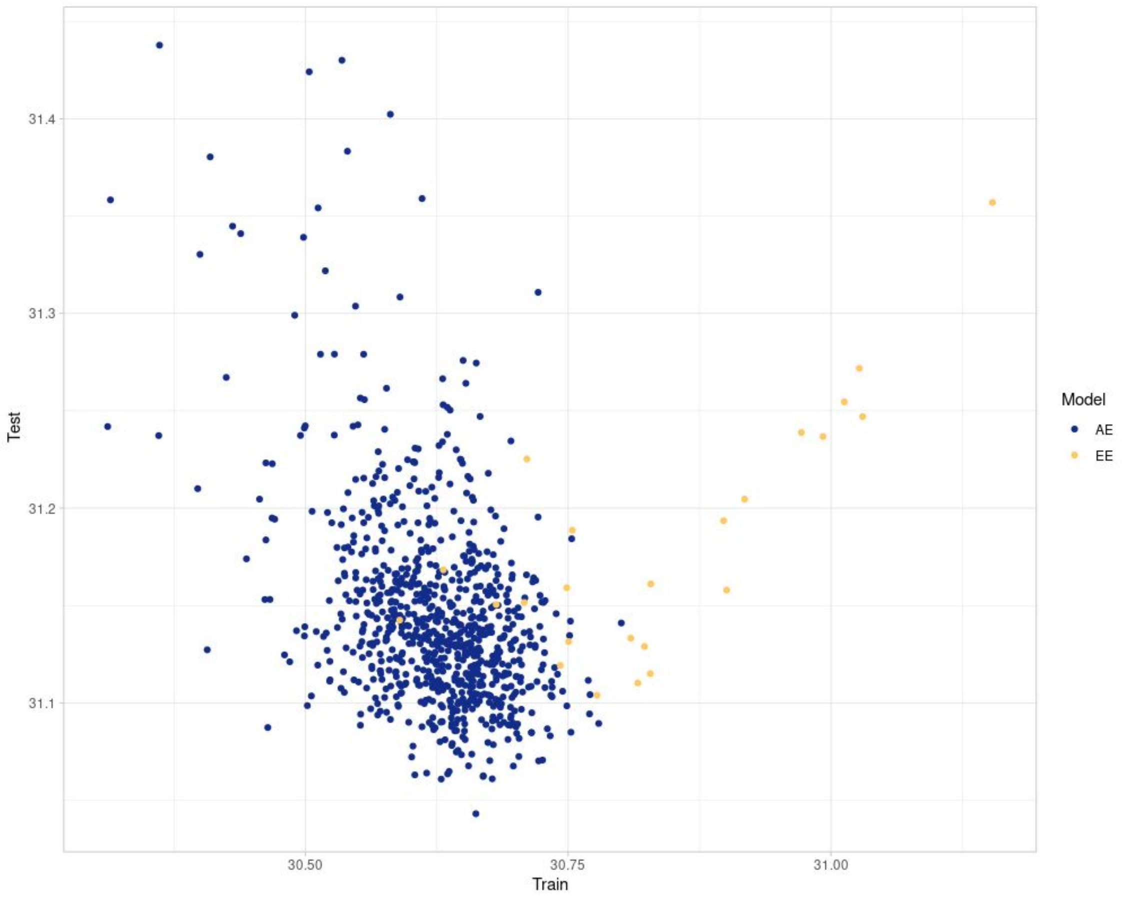
What improvement comes with adding autoencoder?

AE	Scaled weights	Loss function	Difference
1	No	30,674	
1	Yes	30,380	0,294

What improvement comes with adding autoencoder?

AE	Scaled weights	Neurons	Loss function	Difference
2	Yes	8	30,348	
2	Yes	11	30,336	0,012

Results on train and test sets - loss functions



Bibliography

Wuthrich, Mario. (2019). From Generalized Linear Models to Neural Networks, and Back. SSRN Electronic Journal.

Erhan, Dumitru and Manzagol, Pierre-Antoine and Bengio, Y. and Bengio, S. and Vincent, Pascal . (2009). The Difficulty of Training Deep Architectures and the Effect of Unsupervised Pre-Training. Twelfth International Conference on Artificial Intelligence and Statistics

Vincent, Pascal and Larochelle, Hugo and Lajoie, Isabelle and Bengio, Y. and Manzagol, Pierre-Antoine. (2008). Extracting and composing robust features with denoising autoencoders. In ICML 2008: Proceedings of the Twenty-fifth International Conference on Machine Learning

Vincent, Pascal and Larochelle, Hugo and Lajoie, Isabelle and Bengio, Y. and Manzagol, Pierre-Antoine. (2010). Stacked Denoising Autoencoders: Learning Useful Representations in a Deep Network with a Local Denoising Criterion., Journal of Machine Learning Research

Contact

Anna Kozak
anna.kozak@quantee.ai
in kozakanna

Łukasz Delong
lukasz.delong@quantee.ai
lukasz.delong@sgh.waw.pl

Thank you for your attention!