Institute and Faculty of Actuaries

Towards Machine Learning: AI as an alternative to GLM's for Ratemaking

Navarun Jain
Lux Actuaries & Consultants
Dubai, UAE

23 October 2018

Agenda

Overview of Machine Learning

- Introduction to Machine Learning
- Types of Algorithms

Generalized Linear Models

- Poisson – Gamma GLM's
- Tweedie Compound-Poisson GLM's

Artificial Neural Networks

- Structure and Architecture
- How ANN's Work and Learn

Applications to Insurance Data

- Fitting models
- Comparison of Poisson-Gamma GLM, Tweedie GLM and Neural Network on data

Building and Training Neural Networks

Key Takeaways and Conclusions

Institute and Faculty of Actuaries

Machine Learning

23 October 2018

Agenda

Overview of Machine Learning

- Introduction to Machine Learning
- Types of Algorithms

Generalized Linear Models

- Poisson – Gamma GLM's
- Tweedie Compound-Poisson GLM's

Artificial Neural Networks

- Structure and Architecture
- How ANN's Work and Learn

Applications to Insurance Data

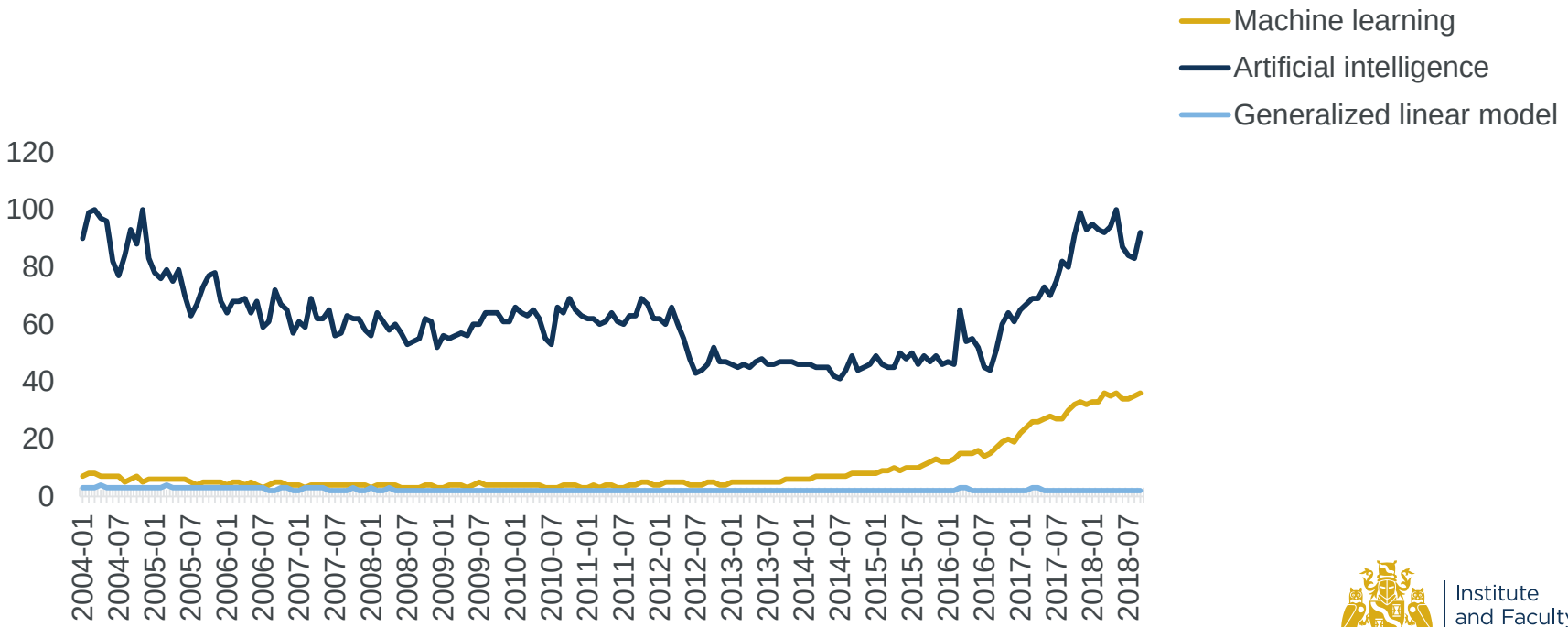
- Fitting models
- Comparison of Poisson-Gamma GLM, Tweedie GLM and Neural Network on data

Building and Training Neural Networks

Key Takeaways and Conclusions

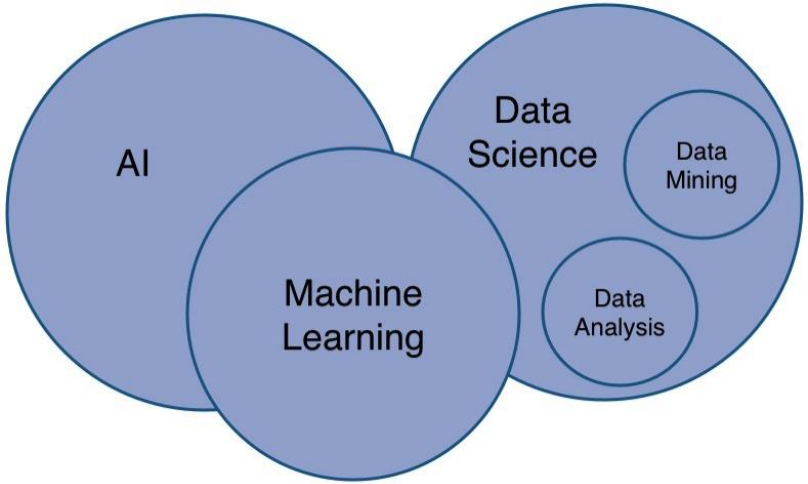
Institute and Faculty of Actuaries

Machine Learning in Perspective



Institute and Faculty of Actuaries

Machine Learning



Institute and Faculty of Actuaries

Machine Learning

The ”teaching a kid math” analogy

Institute and Faculty of Actuaries

Machine Learning

All about patterns!!!

Institute and Faculty of Actuaries

The Roadmap

All about patterns!!!

Computer systems learn from data

We **train** system → System **learns** from that → Then performs operations **on its own**

Institute and Faculty of Actuaries

The Roadmap

All about patterns!!!

Computer systems learn from data

We **train** system → System **learns** from that → Then performs operations **on its own**

Training phase 1: data is fed into the algorithm, relevant fields and records sorted from data to retrieve **active dataset**

Institute and Faculty of Actuaries

The Roadmap

All about patterns!!!

Computer systems learn from data

We **train** system → System **learns** from that → Then performs operations **on its own**

Training phase 2: **Model Fitting** – algorithm decodes hidden patterns and relationships in the data

Institute and Faculty of Actuaries

The Roadmap

All about patterns!!!

Computer systems learn from data

We **train** system → System **learns** from that → Then performs operations **on its own**

Testing phase: new data fed into system, algorithm uses patterns & relationships learnt during the training phase to predict new cases

Institute and Faculty of Actuaries

Types of Algorithms

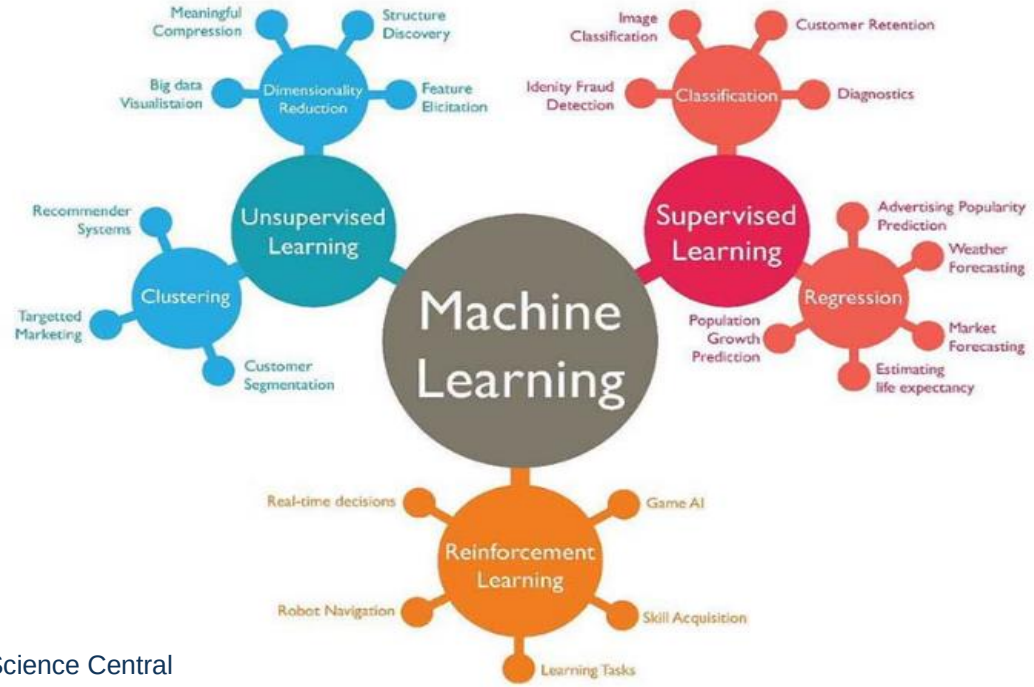


Image Source: Data Science Central

Institute and Faculty of Actuaries

Generalized Linear Models

23 October 2018

Agenda

Overview of Machine Learning

- Introduction to Machine Learning
- Types of Algorithms

Generalized Linear Models

- Poisson – Gamma GLM's
- Tweedie Compound-Poisson GLM's

Artificial Neural Networks

- Structure and Architecture
- How ANN's Work and Learn

Applications to Insurance Data

- Fitting models
- Comparison of Poisson-Gamma GLM, Tweedie GLM and Neural Network on data

Building and Training Neural Networks

Key Takeaways and Conclusions

Institute and Faculty of Actuaries

Generalized Linear Models

$$y_i = \sum_{j=1}^{n} \alpha_j x_{ij} + \epsilon_i$$ ← Classical linear model

$$D = \sum(y - \mu)^2$$ ← Error function

Components of this model:

- Random component – Specifies distribution of target variable
- Systematic component – covariates (x's) produce linear predictor (exp.1)
- Systematic component linked to random component

  g(μ) = η → **Link Function**

$$\eta = \sum_{j=1}^{n} \alpha_j x_j$$

Exp. 1

Institute and Faculty of Actuaries

Poisson-Gamma GLM's

- Claim frequency follows Poisson model with Log link
- Claim severity follows Gamma model with Log link

$$Expected\ Risk\ Premium = Expected\ Claim\ Frequency \times Expected\ Claim\ Severity$$

Poisson model

Gamma model

Institute and Faculty of Actuaries

Tweedie Compound Poisson GLM's

Standard GLM's – Random Component comes from Exponential Family

Distribution	Variance
Gaussian	ϕ
Binomial	$\frac{\mu_i(1-\mu_i)}{n_i}$
Poisson	μ_i
Gamma	$\phi\mu_i^2$
Inverse Gaussian	$\phi\mu_i^3$

Institute and Faculty of Actuaries

Tweedie Compound Poisson GLM's

Tweedie models – Variance has a more general relation with expected value

$$V(\mu) = \mu^p$$

p can take any value and is called the *variance power parameter*

Tweedie Compound Poisson GLM's

Tweedie models – Variance has a more general relation with expected value

$$V(\mu) = \mu^p$$

p can take any value and is called the *variance power parameter*

Compound Poisson – p between 1 and 2

Institute and Faculty of Actuaries

Tweedie Compound Poisson GLM's

Basic idea behind CP models in Ratemaking

- N_i - observed claim count for *i*th category; Z_i - observed claim cost for that category (assuming 1 exposure year)
- N_i ~ Pois(λ_i) & Z_i ~ Gamma(τ_i, α)
- $Z_i | N_i$ ~ Gamma w/mean $N_i \tau_i$
- Tweedie approach assumes: $\mu_i = E(Y_i) = \lambda_i \tau_i$ & $V(Y_i) = \phi \mu_i^p$ where

$$p = \frac{\alpha + 2}{\alpha + 1}$$

- Since $\alpha > 0$, we must have $\mathbf{1 < p < 2}$
- For 0 claims, distribution approximates to Poisson; for non-zero claims, approximates towards Gamma

Institute and Faculty of Actuaries

The Big Questions

The Big Questions

- What if claims follow a fit different from Poisson-Gamma?

The Big Questions

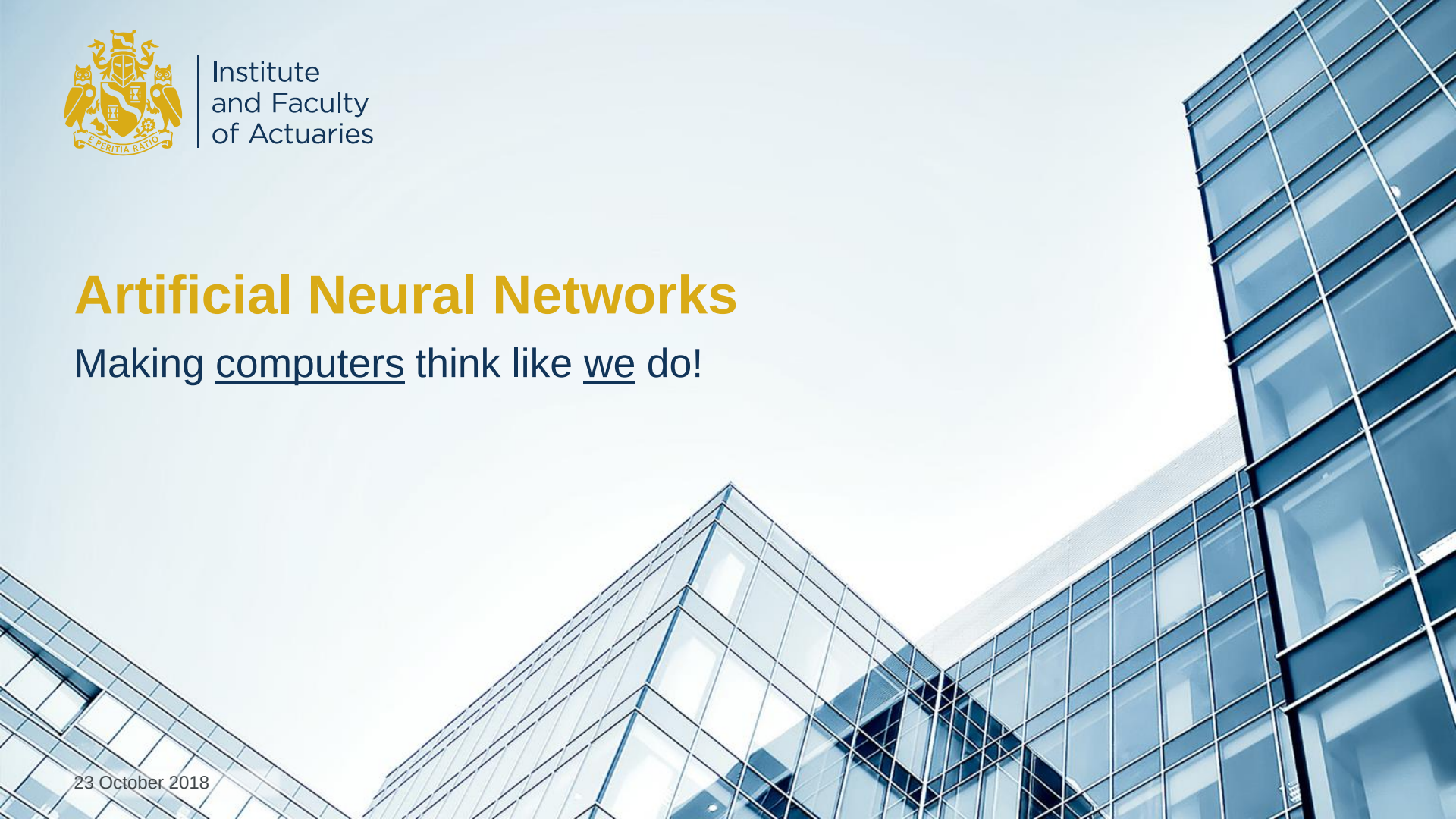
- What if claims follow a fit different from Poisson-Gamma?
- What if we didn't have to look for a fit at all?

Institute and Faculty of Actuaries

The Big Questions

- What if claims follow a fit different from Poisson-Gamma?
- What if we didn't have to look for a fit at all?
- What if there was a way to be able to automate pure premium modelling and potentially find a better fit at the same time?

Institute and Faculty of Actuaries



Institute and Faculty of Actuaries

Artificial Neural Networks

Making computers think like we do!

23 October 2018

Agenda

Overview of Machine Learning

- Introduction to Machine Learning
- Types of Algorithms

Generalized Linear Models

- Poisson – Gamma GLM's
- Tweedie Compound-Poisson GLM's

Artificial Neural Networks

- Structure and Architecture
- How ANN's Work and Learn

Applications to Insurance Data

- Fitting models
- Comparison of Poisson-Gamma GLM, Tweedie GLM and Neural Network on data

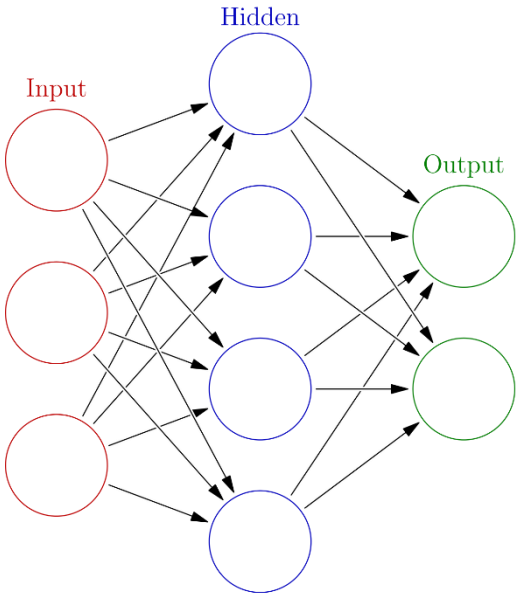
Building and Training Neural Networks

Key Takeaways and Conclusions

Institute and Faculty of Actuaries

Artificial Neural Networks

Structured Sequential model



Structured: A Neural Network has a defined structure that consists of 3 types of layers

Sequential: Information flows in a sequence from one layer to the next, undergoing operations at each layer – almost like an assembly line

Institute and Faculty of Actuaries

How ANN's work

How ANN's work

- Data in every neuron is transformed by an activation function:

$$h_k(x) = g(\beta_{0k} + \sum_{i=1}^{n} x_i \beta_{ik})$$

$h_k(x)$ – k^{th} neuron in a hidden layer
β_{ik} - coefficient of the i^{th} previous-layer neuron on above neuron

Institute and Faculty of Actuaries

How ANN's work

- Data in every neuron is transformed by an activation function:

$$h_k(x) = g(\beta_{0k} + \sum_{i=1}^{n} x_i \beta_{ik})$$

$h_k(x)$ – k^{th} neuron in a hidden layer
β_{ik} - coefficient of the i^{th} previous-layer neuron on above neuron

- Activation function transforms the linear combination of inputs from one layer and sends it to the next layer.

Institute and Faculty of Actuaries

How ANN's Learn

How ANN's Learn

- At first, each neuron is randomly assigned a weight – this measures the contribution of that neuron to the next layer.

How ANN's Learn

- At first, each neuron is randomly assigned a weight – this measures the contribution of that neuron to the next layer.
- Data flows through network, predicted values calculated.

Institute and Faculty of Actuaries

How ANN's Learn

- At first, each neuron is randomly assigned a weight – this measures the contribution of that neuron to the next layer.
- Data flows through network, predicted values calculated.
- Predictions are compared with actuals based on a loss function.

Institute and Faculty of Actuaries

How ANN's Learn

- At first, each neuron is randomly assigned a weight – this measures the contribution of that neuron to the next layer.
- Data flows through network, predicted values calculated.
- Predictions are compared with actuals based on a loss function.
- Weights are updated to reduce value of loss function.

Institute and Faculty of Actuaries

Optimizing Neural Networks

Optimizing Neural Networks

- Learning continues until the following is minimized:

Optimizing Neural Networks

- Learning continues until the following is minimized:

$$\nabla_{\mathrm{W}}\mathrm{L} = \frac{\delta L}{\delta W}$$

Gradient of the Loss function – measures change in loss function as model weights change

Optimizing Neural Networks

- Learning continues until the following is minimized:

$$\nabla_{\mathrm{W}}\mathrm{L} = \frac{\delta L}{\delta W}$$

Gradient of the Loss function – measures change in loss function as model weights change

- The above function is computed and a step is taken in the direction where it is minimized the most.

Institute and Faculty of Actuaries

Optimizing Neural Networks

- Learning continues until the following is minimized:

$$\nabla_{\mathrm{W}}\mathrm{L} = \frac{\delta L}{\delta W}$$

Gradient of the Loss function – measures change in loss function as model weights change

- The above function is computed and a step is taken in the direction where it is minimized the most.
- Size of this step is the **learning rate.**

Institute and Faculty of Actuaries

Optimizing Neural Networks

- Suppose for Neuron A and iteration *t*, the weight was found to be W_{At}

Institute and Faculty of Actuaries

Optimizing Neural Networks

- Suppose for Neuron A and iteration *t*, the weight was found to be $W_{A(t)}$
- Then, for iteration *t* + *1*, weight is optimized to:

$$W_{A(t+1)} = W_{A(t)} - \eta \nabla_{W_{A(t)}} L$$

- $\boldsymbol{\eta}$ – Learning Rate
- $\boldsymbol{\nabla_{W_{A(t)}} L}$ – Gradient of Loss Function w.r.t. weight of Neuron A at iteration *t*

Optimizing Neural Networks

- **Vanilla approach: Compute gradient for entire training sample and update weights based on that**

Optimizing Neural Networks

- **Vanilla approach: Compute gradient for entire training sample and update weights based on that**
  - No method to check if full convergence is achieved
  - What if different parameters work differently and require different optimization rates?

Optimizing Neural Networks

- **Vanilla approach: Compute gradient for entire training sample and update weights based on that**
  - No method to check if full convergence is achieved
  - What if different parameters work differently and require different optimization rates?
- **Stochastic Gradient Descent: Compute gradient for each individual point in the training sample and update weights iteratively for every sample**

Institute and Faculty of Actuaries

Optimizing Neural Networks

- **Vanilla approach: Compute gradient for entire training sample and update weights based on that**
  - No method to check if full convergence is achieved
  - What if different parameters work differently and require different optimization rates?
- **Stochastic Gradient Descent: Compute gradient for each individual point in the training sample and update weights iteratively for every sample**
  - Too slow – Might cause algorithm to crash or give up for extremely large datasets, thus potentially preventing full convergence

Institute and Faculty of Actuaries

Optimizing Neural Networks

- In theory, SGD is more accurate

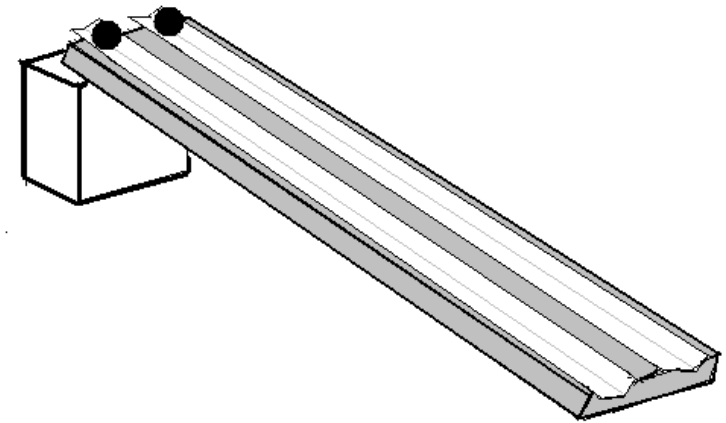
Optimizing Neural Networks

- In theory, SGD is more accurate
- Methods such as **Momentum** and the **Nesterov Accelerated Gradient** improve SGD and make it faster by optimizing learning rates internally

Institute and Faculty of Actuaries

Optimizing Neural Networks

- In theory, SGD is more accurate
- Methods such as **Momentum** and the **Nesterov Accelerated Gradient** improve SGD and make it faster by optimizing learning rates internally



Think of it as a ball rolling down a hill

Institute and Faculty of Actuaries

So what's going on here?

So what's going on here?

The "teaching a kid math" analogy

Institute and Faculty of Actuaries

With ANN's, no need to…

With ANN's, no need to…

- …make assumptions about distributions

With ANN's, no need to…

- …make assumptions about distributions
- …worry about possible correlations between predictors

Institute and Faculty of Actuaries

With ANN's, no need to…

- …make assumptions about distributions
- …worry about possible correlations between predictors
- …look for interactions between predictors

Institute and Faculty of Actuaries

Applications to Insurance Data

dataCar from R's insuranceData package

23 October 2018

Data Description

- Policyholder-level information on one-year vehicle insurance policies
- 67,856 records with following rating factors –
  - Vehicle value in $10,000's
  - Vehicle body type (eg. Sedan, convertible, hatchback, bus & other levels)
  - Vehicle age (Levels 1-4 w/1 being the newest & 4 being the oldest)
  - Gender of driver
  - Area
  - Driver age category (Levels 1-6 w/1 being youngest & 6 being oldest)

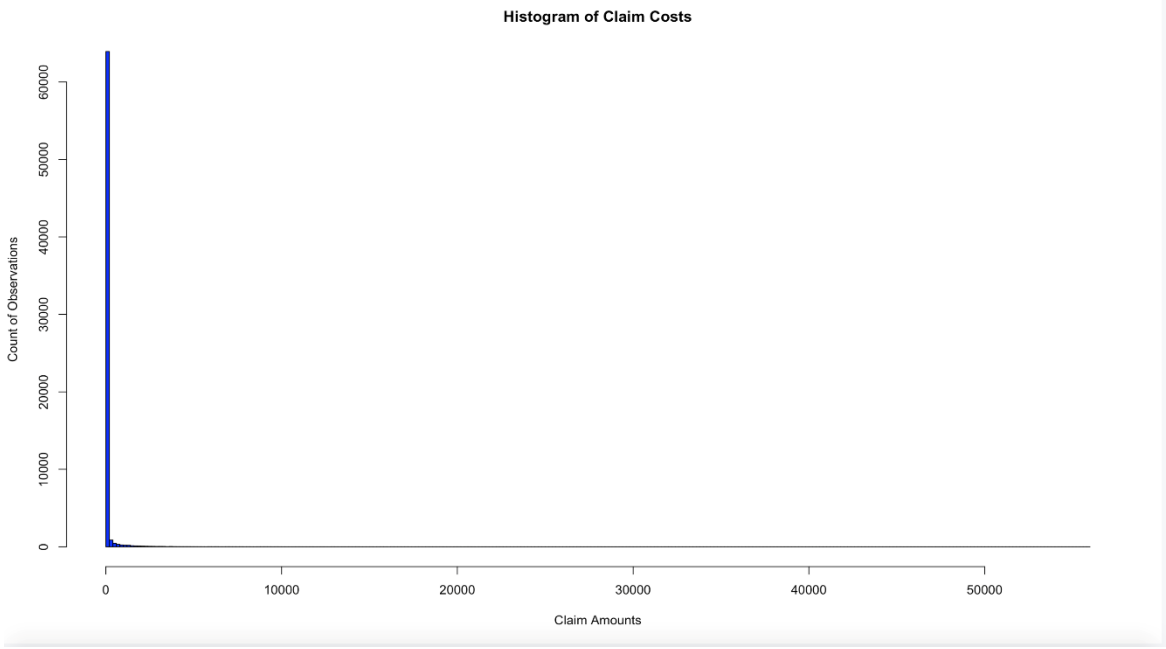
Institute and Faculty of Actuaries

Data Description

- Heavily skewed w/no-claim percentage of 93.2%

Distribution of Claims

- Heavily skewed w/no-claim percentage of 93.2%

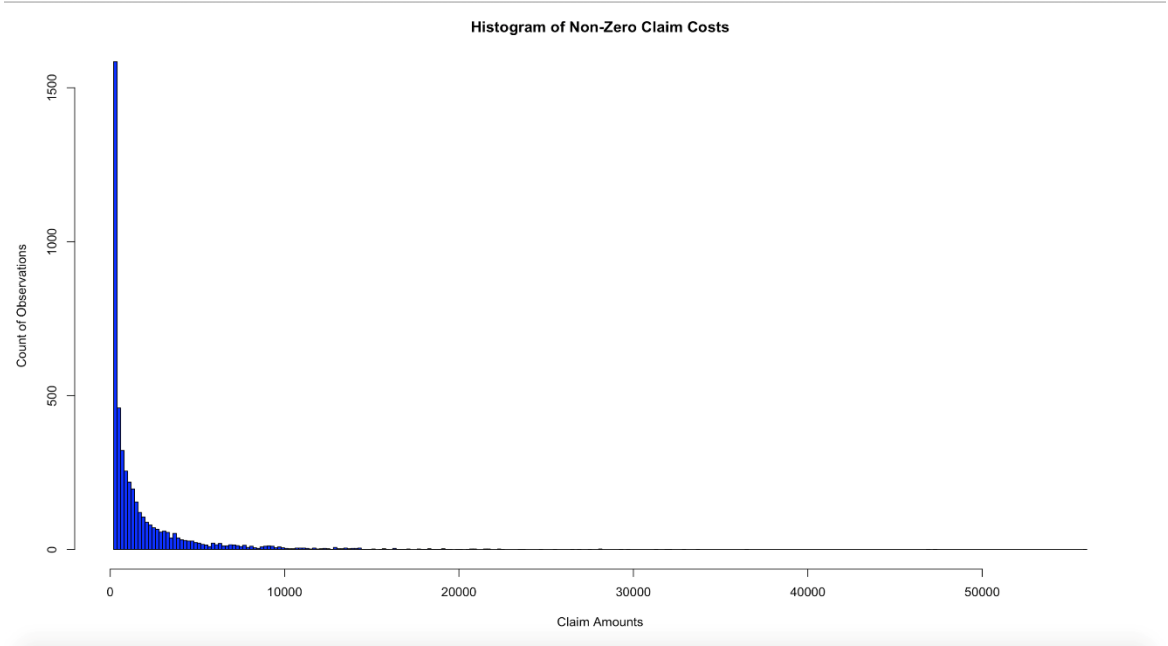


Distribution of raw claims data

Institute and Faculty of Actuaries

Distribution of Claims

- Heavily skewed w/no-claim percentage of 93.2%

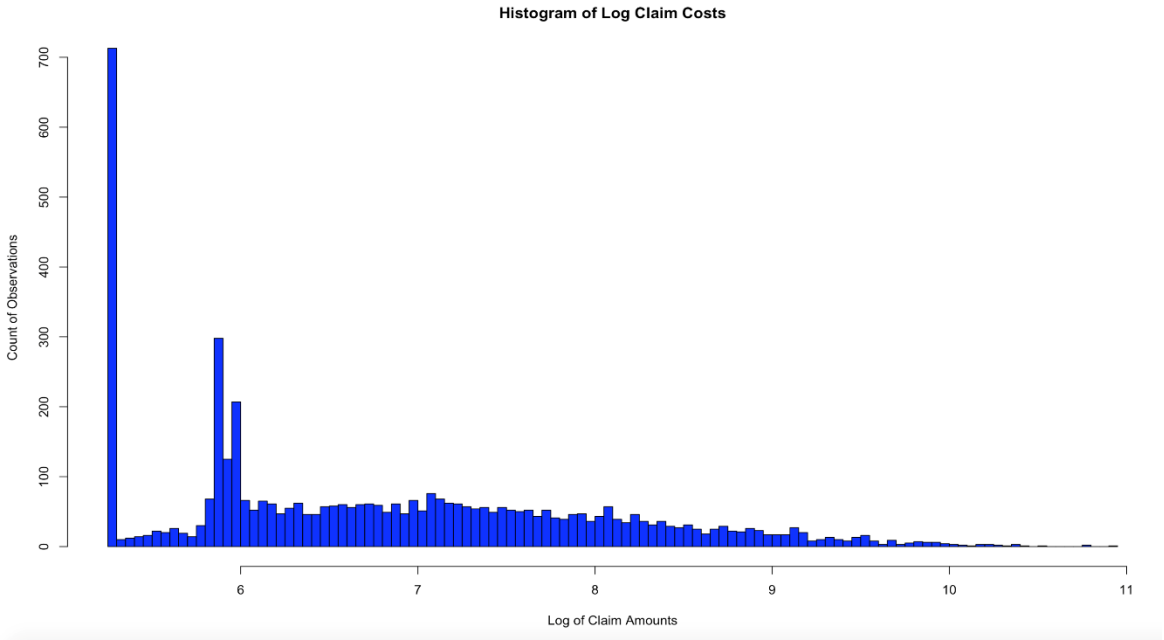


Distribution of
non-zero claims only

Institute and Faculty of Actuaries

Distribution of Claims

- Heavily skewed w/no-claim percentage of 93.2%



Distribution of the logarithm of claims

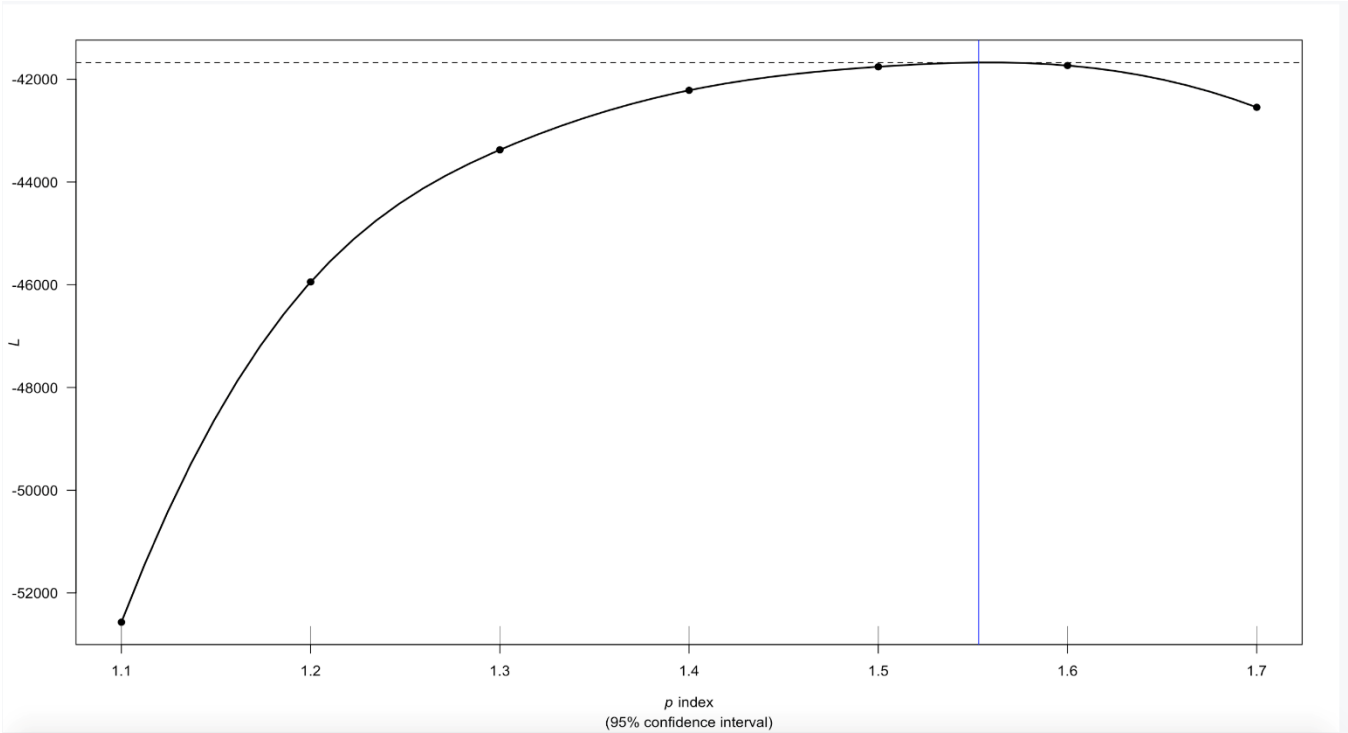
Institute and Faculty of Actuaries

Fitting Tweedie CP GLM

- Optimal variance power parameter tuned by MLE
- Range specified: 1.1 - 1.9 (all values between 1 & 2)
- Optimal value found to be 1.553

Institute and Faculty of Actuaries

Fitting Tweedie CP GLM



Plot of log-likelihood values against p – values. Vertical blue line shows point of maximum likelihood, justifying choice of parameter

Institute and Faculty of Actuaries

Fitting Neural Networks– Choice of Network Architecture

Fitting Neural Networks– Choice of Network Architecture

- Fitted by 5-fold cross-validation
- Base network architectures were trained with the same set of hyperparameters chosen at random

Fitting Neural Networks– Choice of Network Architecture

- Fitted by 5-fold cross-validation
- Base network architectures were trained with the same set of hyperparameters chosen at random
- Architectures chosen: (33-40-1), (33-80-1), (33-100-1), (33-120-1), (33-80-40-1), (33-100-60-1), (33-120-60-1)

Institute and Faculty of Actuaries

Fitting Neural Networks– Choice of Network Architecture

- Fitted by 5-fold cross-validation
- Base network architectures were trained with the same set of hyperparameters chosen at random
- Architectures chosen: (33-40-1), (33-80-1), (33-100-1), (33-120-1), (33-80-40-1), (33-100-60-1), (33-120-60-1)
- Best network architecture found to be (33-120-60-1) – this was chosen for further tuning
- In general, networks with 2 hidden layers fitted better than networks with 1 hidden layer

Institute and Faculty of Actuaries

Fitting Neural Networks – Choosing Hyperparameters

Fitting Neural Networks – Choosing Hyperparameters

- Learning rate and batch size tuned on architecture chosen from previous step
- Tuning done by 5-fold cross-validation
- Following values were chosen to be tested:
  - Learning rate – 0.001, 0.01, 0.05, 0.1
  - Batch size – 3000, 8000, 10000

Institute and Faculty of Actuaries

Fitting Neural Networks – Choosing Hyperparameters

- Learning rate and batch size tuned on architecture chosen from previous step
- Tuning done by 5-fold cross-validation
- Following values were chosen to be tested:
  - Learning rate – 0.001, 0.01, 0.05, 0.1
  - Batch size – 3000, 8000, 10000
- Best hyperparameters found to be
  - Learning rate – 0.1
  - Batch size – 3000

Institute and Faculty of Actuaries

Model Comparison

Model Comparison

- 3 approaches taken
  - Test data predictive accuracy – Test RMSE
  - Resampling Error on Training dataset – 5-fold Cross Validation MSE
  - Goodness-of-fit test – AIC

Test RMSE

Model	Test RMSE ($\times\ 10^3$)
Poisson-Gamma GLM	1.25
Tweedie Compound Poisson GLM	1.25
Artificial Neural Network	1.27

Institute and Faculty of Actuaries

5-Fold Cross-Validation Error

Model	CV MSE
Poisson-Gamma GLM	1.69×10^6
Tweedie Compound Poisson GLM	1.70×10^6
Artificial Neural Network	19.98

Institute and Faculty of Actuaries

AIC

AIC

- Stands for Akaike Information Criterion
- Gives a measure of distance between true and predicted trends

AIC

Model	AIC ($\times 10^5$)
Poisson-Gamma GLM	5.72
Tweedie Compound Poisson GLM	5.73
Artificial Neural Network	5.73

Institute and Faculty of Actuaries

Building and Training Neural Networks

TensorFlow in R

23 October 2018

TensorFlow

Institute and Faculty of Actuaries

TensorFlow

- Developed by Google Brain
- Released in 2015

Institute and Faculty of Actuaries

TensorFlow

- Developed by Google Brain
- Released in 2015
- Can be implemented in
  - Python: TF, Keras
  - R: **Keras**, Estimator, TF API

Institute and Faculty of Actuaries

Building Neural Networks

- Surprisingly easy to build neural networks in R using Keras

Building Neural Networks

- Surprisingly easy to build neural networks in R using Keras
- Models initialized as sequential objects and layers added as nested commands

Building Neural Networks

- Surprisingly easy to build neural networks in R using Keras
- Models initialized as sequential objects and layers added as nested commands
- Once model is constructed, it can be called on a dataset

The Keras Roadmap

The Keras Roadmap

Model Initialization → Layer definition → Compilation & Fitting

An empty neural network is initialized using *keras_model_sequential()*: Layers can then be added sequentially to the model

The Keras Roadmap

Model Initialization → **Layer definition** → Compilation & Fitting

Hidden layers are added and their structure is defined using the command *layer_dense()*: Activation function, # of neurons specified

Institute and Faculty of Actuaries

The Keras Roadmap

Model Initialization → Layer definition → **Compilation & Fitting**

After model architecture is fixed, hyperparameters, optimizer and loss function are set and using the command *compile()*. Following this, model is trained using the command *fit()*.

Try it out!!!

Institute and Faculty of Actuaries

Key Takeaways & Conclusions

23 October 2018

AI: The Good and the Not-so-good

Institute and Faculty of Actuaries

AI: The Good and the Not-so-good

- The Good:
  - Allows for complete automation
  - No need to assume anything about the data, both in terms of rating factors and claim distributions

AI: The Good and the Not-so-good

- The Good:
  - Allows for complete automation
  - No need to assume anything about the data, both in terms of rating factors and claim distributions
- The Not-so-good:
  - Computationally intensive – requires hardware such as GPU's and fast/powerful processors to run efficiently
  - Interpretability

Institute and Faculty of Actuaries

Conclusions

Conclusions

- Machine Learning and AI are powerful tools, can aid actuaries in decision-making
- AI should definitely be explored and experimented with in addition to using GLM's

Institute and Faculty of Actuaries

Conclusions

- Machine Learning and AI are powerful tools, can aid actuaries in decision-making
- AI should definitely be explored and experimented with in addition to using GLM's
- No one "right" model – best predictions can come from ensemble models

Institute and Faculty of Actuaries

Conclusions

- Machine Learning and AI are powerful tools, can aid actuaries in decision-making
- AI should definitely be explored and experimented with in addition to using GLM's
- No one "right" model – best predictions can come from ensemble models
- Further research being done to improve interpretability of AI, applications of Machine Learning in the actuarial realm

Institute and Faculty of Actuaries

Selected References & Suggested Reading

- Fox, J., *Applied Regression Analysis & Generalized Linear Models (Third Edition)*, Sage Publications, 2016
- Andersen, D. et al, *A Practitioner's Guide to Generalized Linear Models: A foundation for theory, interpretation and application*, in CAS 2004 Discussion Paper Program
- Smyth, G.K., Jørgensen, B., *Fitting Tweedie's Compound Poisson Model to Insurance Claims Data: Dispersion Modelling*, in ASTIN Bulletin, Vol. 32, No. 1, 2002, pp. 143 – 157
- Rumelhart, D. E., Hinton, G. E., Williams, R. J., *Learning representations by back-propagating errors, in Nature Vol. 323*, October 1986
- TensorFlow: *www.tensorflow.org*
- The R Interface to Keras: *https://keras.rstudio.com/*
- Léon Bottou, *Large-Scale Machine Learning with Stochastic Gradient Descent*: http://khalilghorbal.info/assets/spa/papers/ML_GradDescent.pdf

Institute and Faculty of Actuaries

Questions

Comments

Institute and Faculty of Actuaries