Society of Actuaries in Ireland

Introduction to Artificial Intelligence

8 April 2019

Disclaimer

The views expressed in this presentation are those of the presenter(s) and not necessarily those of the Society of Actuaries in Ireland or their employers.

Welcome

- Conor Byrne
  - Deputy Chair, SAI Data Analytics Subcommittee

Agenda

- What is AI?
- Regression/Classification vs Specification
- How do Neural Networks work?
- Gradient Descent Optimisation
- Why Should Actuaries be Interested?
- Examples

What is AI?

- AI is where the machine's actions/output is indistinguishable from a trained person's actions/output

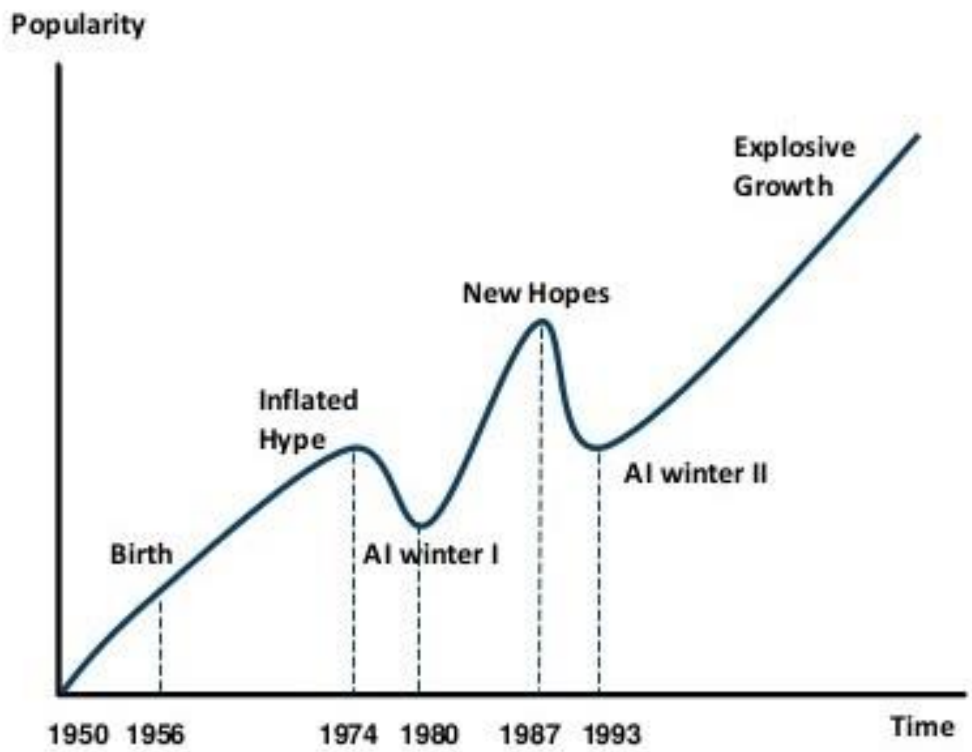
- Types of AI:
  - Artificial General Intelligence
  - Artificial Narrow Intelligence

General Examples

Self-Driving Cars	Speech-to-text	Fraud Detection	Customer Retention
Game Playing	Machine translation	Pricing	Proxy Models
Reducing Electricity Costs	Chatbots	Credit Risk	Call-Centre Routing
Analysing Satellite Photos	Recommender Systems	Sales Forecasting	Sentiment Analysis
Reading X-rays	Text-to-Speech	Anti-Money Laundering	Geographic Analysis

History of AI

AI HAS A LONG HISTORY OF BEING "THE NEXT BIG THING"...



Timeline of AI Development

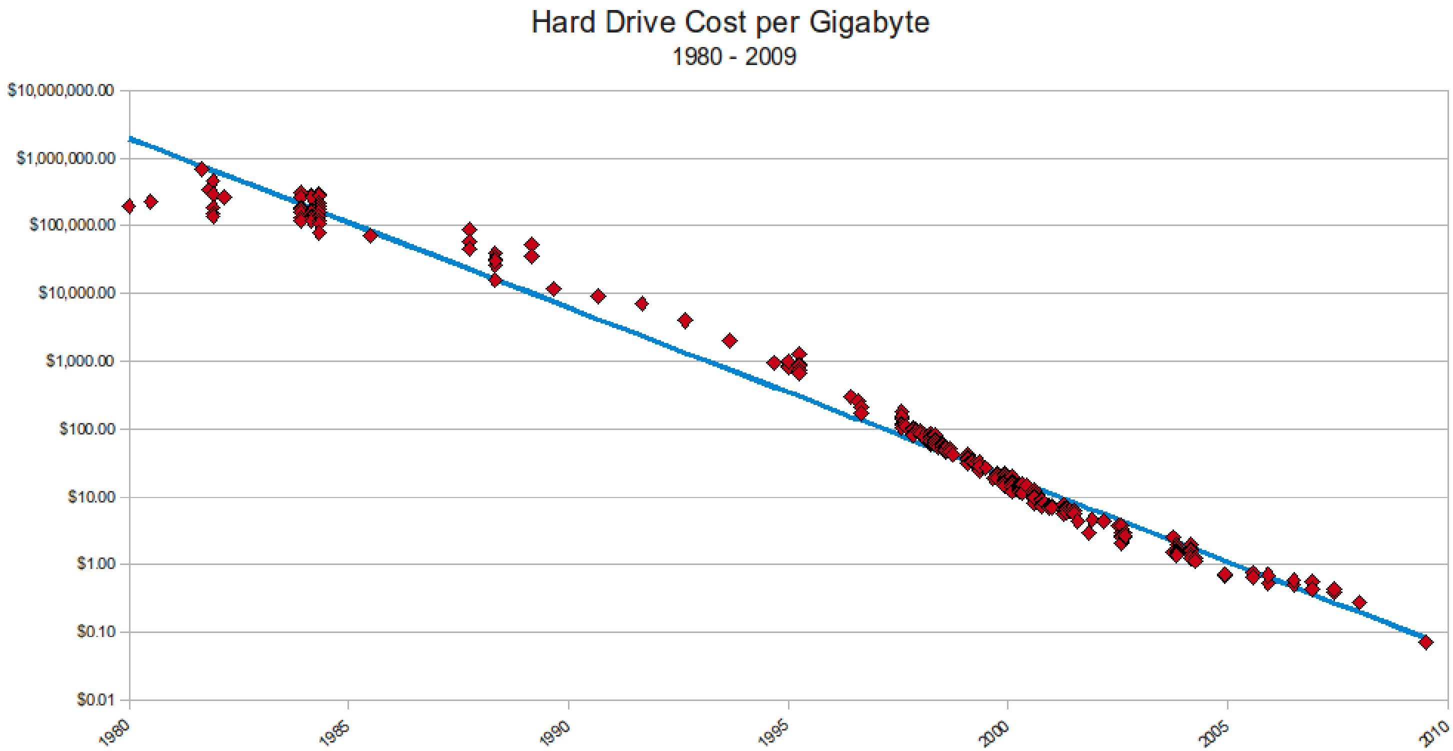
- **1950s-1960s**: First AI boom - the age of reasoning, prototype AI developed
- **1970s**: AI winter I
- **1980s-1990s**: Second AI boom: the age of Knowledge representation (appearance of expert systems capable of reproducing human decision-making)
- **1990s**: AI winter II
- **1997**: Deep Blue beats Gary Kasparov
- **2006**: University of Toronto develops Deep Learning
- **2011**: IBM's Watson won Jeopardy
- **2016**: Go software based on Deep Learning beats world's champions

https://www.actuaries.digital/2018/09/05/history-of-ai-winters/

Types of AI

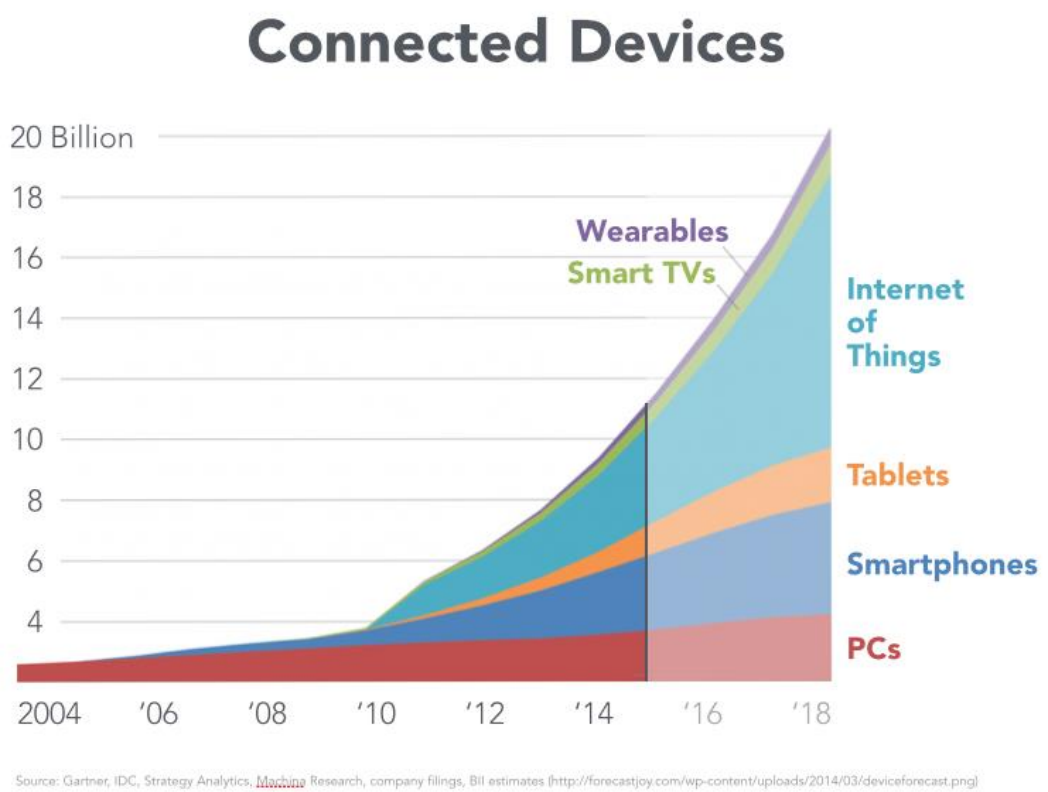
- Tribes of AI
  - Connectionists (inspired by neuroscience)
  - Bayesians (learn from experience)
  - Evolutionists (inspired by evolution)
  - Symbolists (if….Then…elseif….then….therefore)
  - Analogisers (Learn new things based on existing knowledge base)

Data Storage Costs

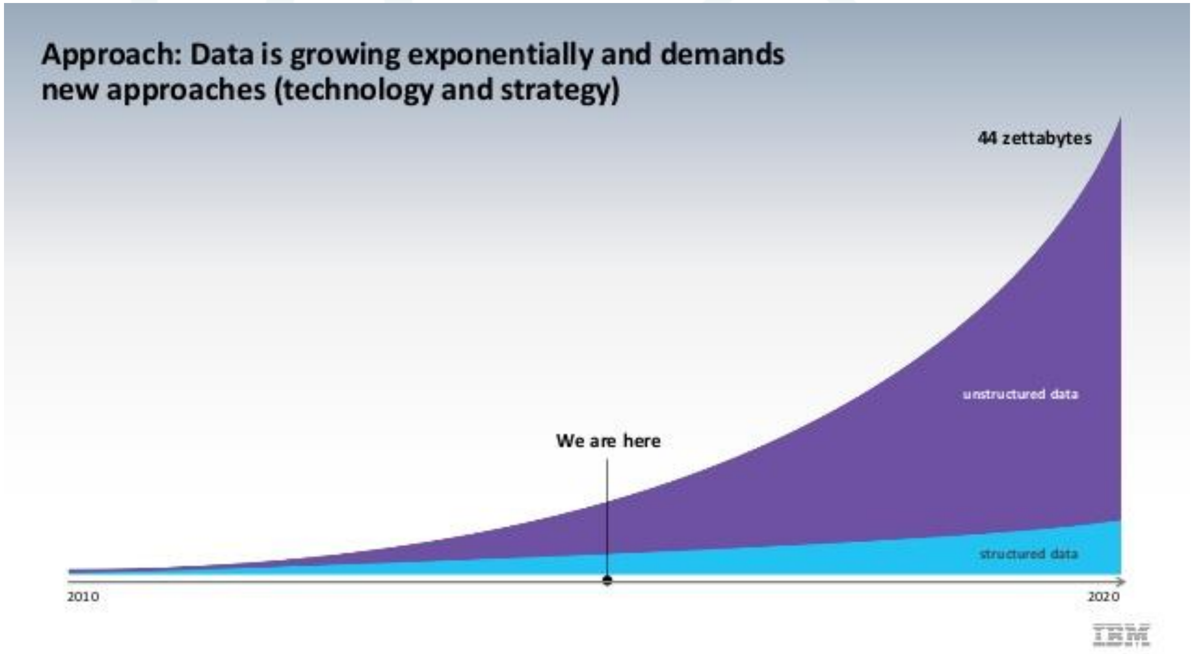


Digitalization

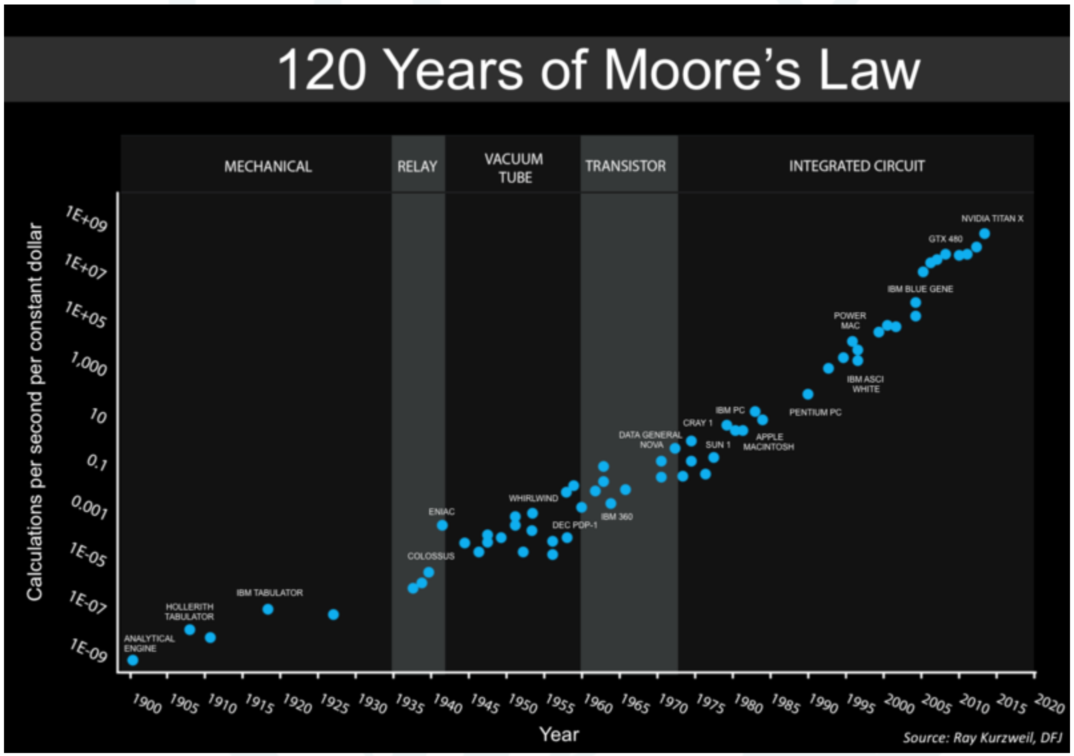
Number of Wifi-Connected Devices



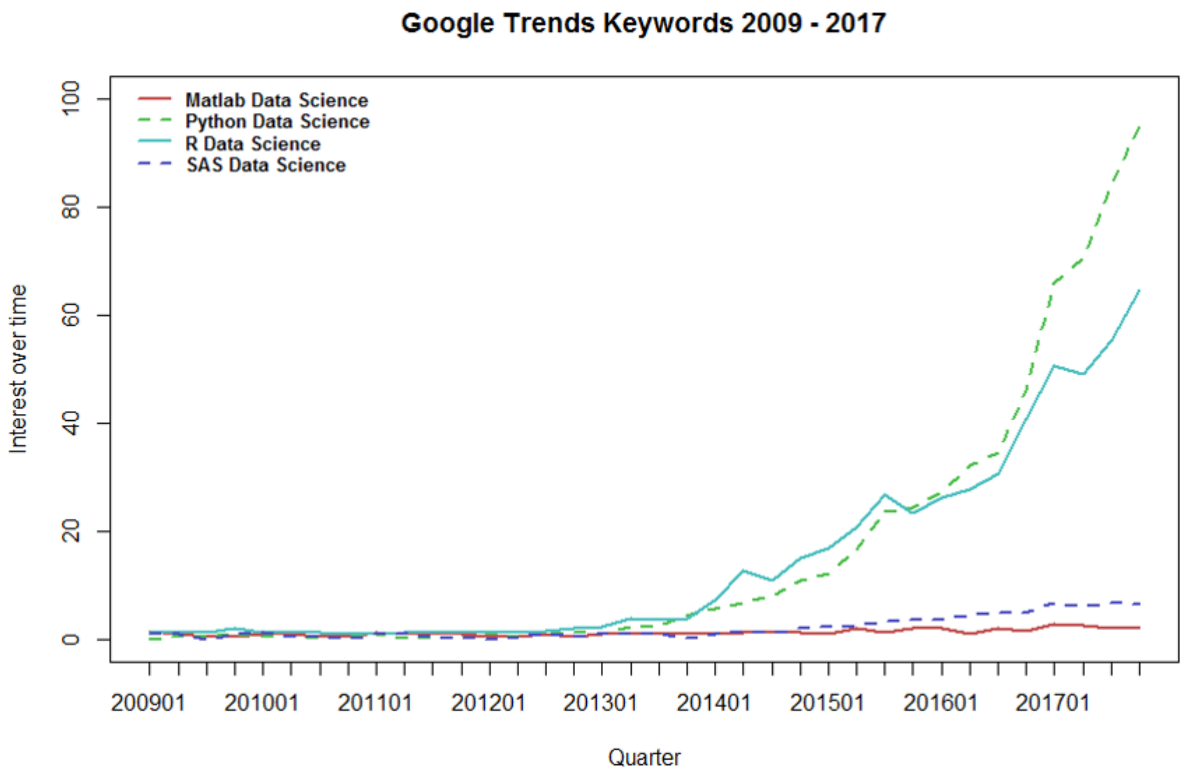
Volume of Data



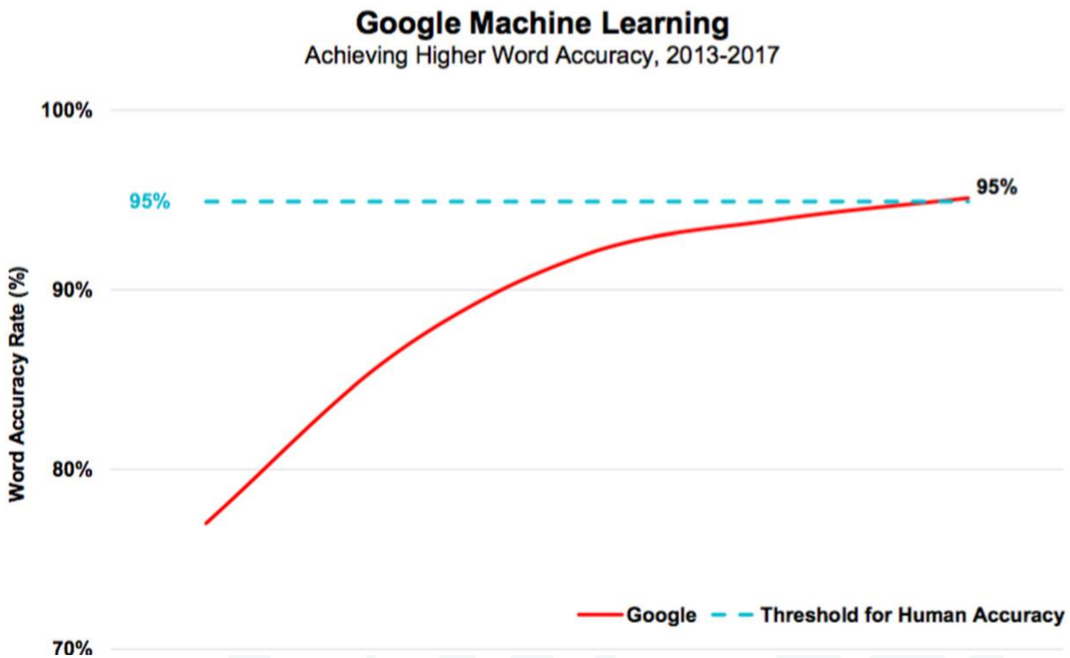
Computer Speeds



Data Science Tools



Machine Learning



Agenda

- What is AI?
- Regression/Classification vs Specification
- How do Neural Networks work?
- Gradient Descent Optimisation
- Why Should Actuaries be Interested?
- Examples

What are Neural Networks Used For?

- Regression
  - Predicting a real number
- Classification
  - Predicting what category something belongs to
- Unsupervised Learning
  - E.g. Clustering

Regression/Classification vs Specification

Functional Specification:

- Define every single step in the process
- Then implement each step

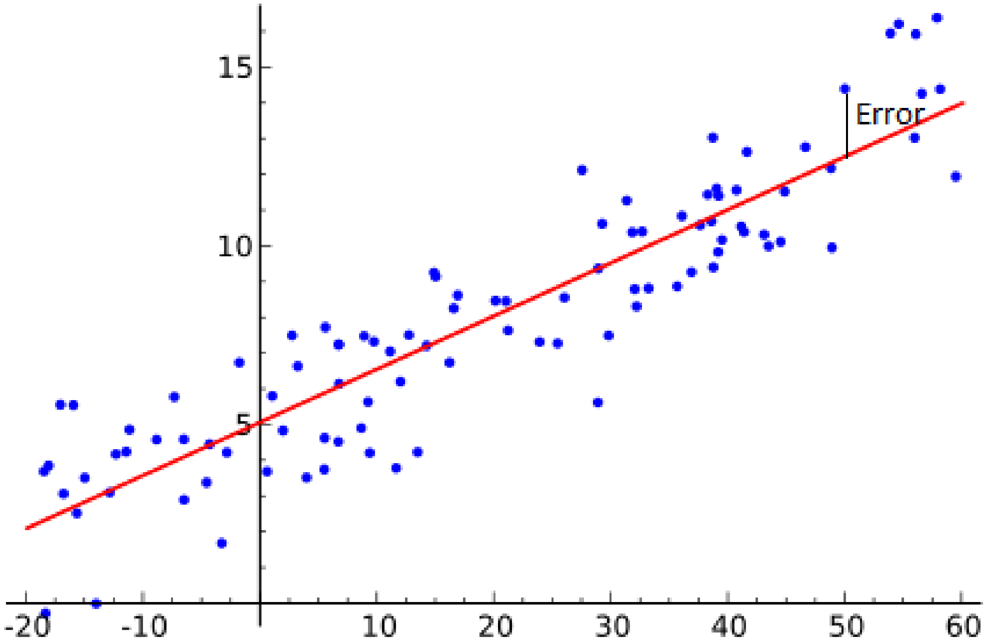
Regression/Classification

- Define the architecture of the model
- Tell the model what the output should be
- Let the computer find the optimal model
  - Which gives the best match to the desired output

Regression

- Linear Regression Model:

$$\hat{Y} = b + aX$$



- Choose Loss Function

  (e.g. Sum of Square Errors)

- Choose parameters a and b which minimise the loss function

- Neural Network Model:

$$\hat{Y} = f_1(b_1 + a_1 * f_2(b_2 + a_2 * f_3(\ldots\ldots\ldots f_n(b_n + a_n X))))$$

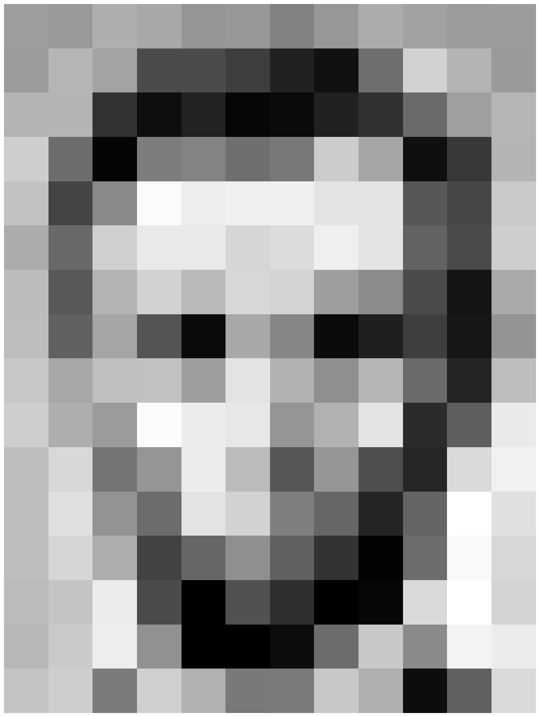
Classification

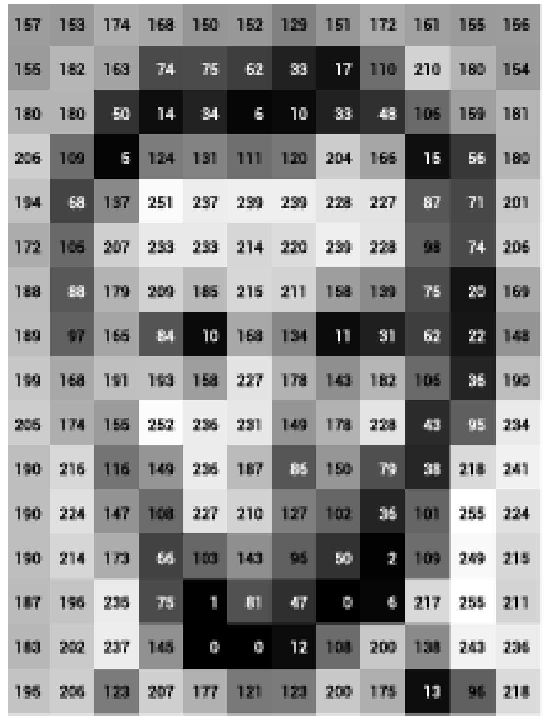
$f_1($  $) =$ "cat"

$f_1($  $) =$ "dog"

https://www.csie.ntu.edu.tw/~yvchen/doc/TSMC_ML-Tutorial.pdf

Digital Photos





157	153	174	168	150	152	129	151	172	161	155	156
155	182	163	74	75	62	33	17	110	210	180	154
180	180	50	14	34	6	10	33	48	106	159	181
206	109	5	124	131	111	120	204	166	15	56	180
194	68	137	251	237	239	239	228	227	87	71	201
172	105	207	233	233	214	220	239	228	98	74	206
188	88	179	209	185	215	211	158	139	75	20	169
189	97	165	84	10	168	134	11	31	62	22	148
199	168	191	193	158	227	178	143	182	106	36	190
205	174	155	252	236	231	149	178	228	43	95	234
190	216	116	149	236	187	86	150	79	38	218	241
190	224	147	108	227	210	127	102	36	101	255	224
190	214	173	66	103	143	96	50	2	109	249	215
187	196	235	75	1	81	47	0	6	217	255	211
183	202	237	145	0	0	12	108	200	138	243	236
195	206	123	207	177	121	123	200	175	13	96	218

- Digital Photos are stored as arrays of numbers

Source: Openframeworks.cc

Classification

$f_1($

50	14	34	6	10	33	48	106	159	181
5	124	131	111	120	204	166	15	56	180
137	251	237	239	239	228	227	87	71	201
207	233	233	214	220	239	228	98	74	206
179	209	185	215	211	158	139	75	20	169
165	84	10	168	134	11	31	62	22	148
191	193	158	227	178	143	182	106	36	190
155	252	236	231	149	178	228	43	95	234
116	149	236	187	86	150	79	38	218	241
147	108	227	210	127	102	36	101	255	224
173	66	103	143	96	50	2	109	249	215
235	75	1	81	47	0	6	217	255	211

$) =$ "cat"

$f_1($

180	180	50	14	34	6	10	33	48	106	159
206	109	5	124	131	111	120	204	166	15	56
194	68	137	251	237	239	239	228	227	87	71
172	105	207	233	233	214	220	239	228	98	74
188	88	179	209	185	215	211	158	139	75	20
189	97	165	84	10	168	134	11	31	62	22
199	168	191	193	158	227	178	143	182	106	36
205	174	155	252	236	231	149	178	228	43	95
190	216	116	149	236	187	86	150	79	38	218
190	224	147	108	227	210	127	102	36	101	255
190	214	173	66	103	143	96	50	2	109	249
187	196	235	75	1	81	47	0	6	217	255
183	202	237	145	0	0	12	108	200	138	243

$) =$ "dog"

https://www.csie.ntu.edu.tw/~yvchen/doc/TSMC_ML-Tutorial.pdf

Digital Text

- Can be converted to vectors of numbers
  - Glove
  - Word2Vec
  - Word Embeddings

Classification

$f_1($

50	14	34	6	10	33	48	106	159	181
5	124	131	111	120	204	166	15	56	180
137	251	237	239	239	228	227	87	71	201
207	233	233	214	220	239	228	98	74	206
179	209	185	215	211	158	139	75	20	169
165	84	10	168	134	11	31	62	22	148
191	193	158	227	178	143	182	106	36	190
155	252	236	231	149	178	228	43	95	234
116	149	236	187	86	150	79	38	218	241
147	108	227	210	127	102	36	101	255	224
173	66	103	143	96	50	2	109	249	215
235	75	1	81	47	0	6	217	255	211

$) =$

1	0	0	0	0

$f_1($

180	180	50	14	34	6	10	33	48	106	159
206	109	5	124	131	111	120	204	166	15	56
194	68	137	251	237	239	239	228	227	87	71
172	105	207	233	233	214	220	239	228	98	74
188	88	179	209	185	215	211	158	139	75	20
189	97	165	84	10	168	134	11	31	62	22
199	168	191	193	158	227	178	143	182	106	36
205	174	155	252	236	231	149	178	228	43	95
190	216	116	149	236	187	86	150	79	38	218
190	224	147	108	227	210	127	102	36	101	255
190	214	173	66	103	143	96	50	2	109	249
187	196	235	75	1	81	47	0	6	217	255
183	202	237	145	0	0	12	108	200	138	243

$) =$

0	1	0	0	0

$$\hat{Y} = f_1(b_1 + a_1 * f_2(b_2 + a_2 * f_3(\ldots\ldots\ldots f_n(b_n + a_n X)))$$

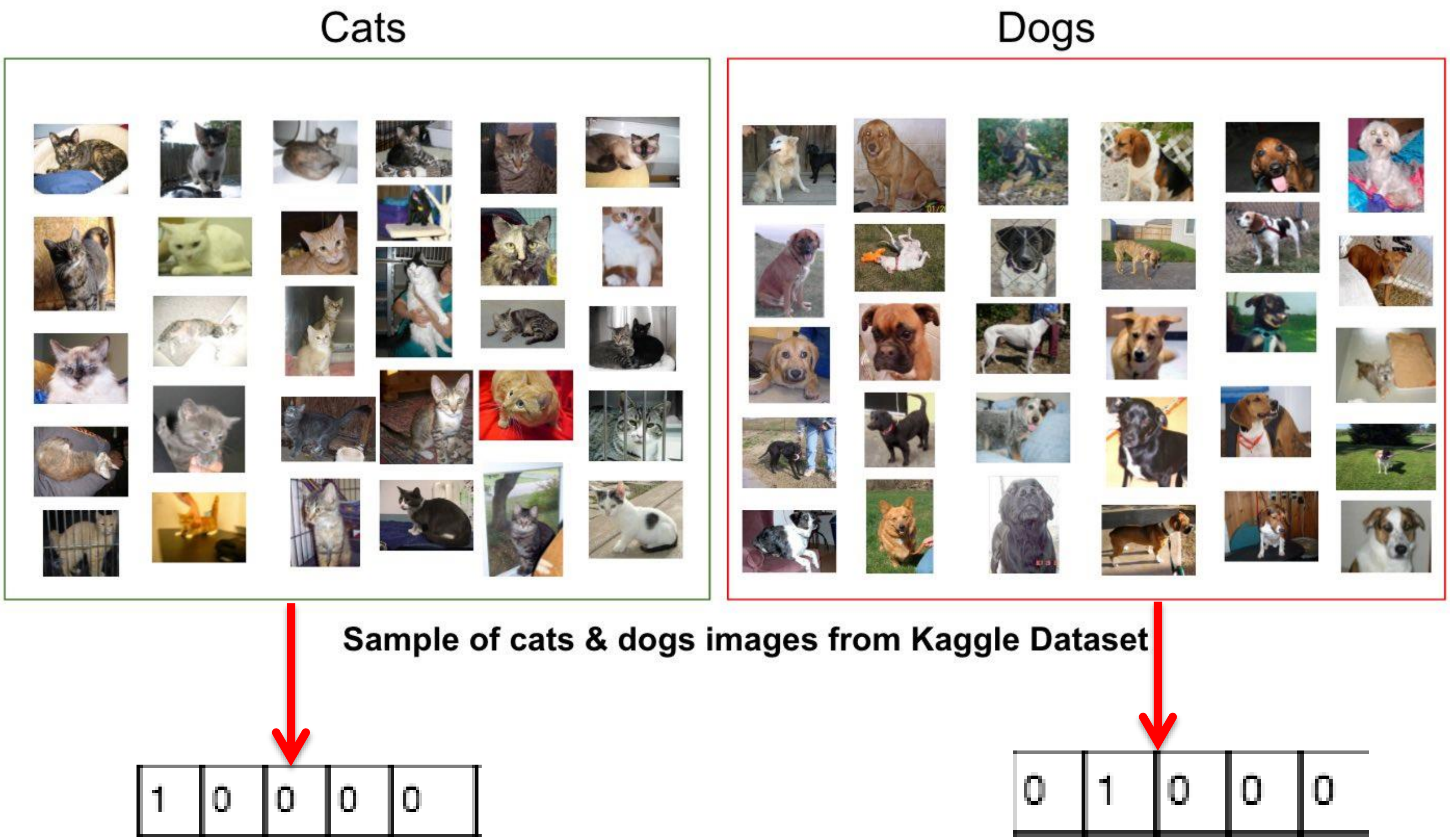
Classification

$f_1($  $) =$ "cat"

$f_1($  $) =$ "dog"

https://www.csie.ntu.edu.tw/~yvchen/doc/TSMC_ML-Tutorial.pdf

Classification



Regression and Classification

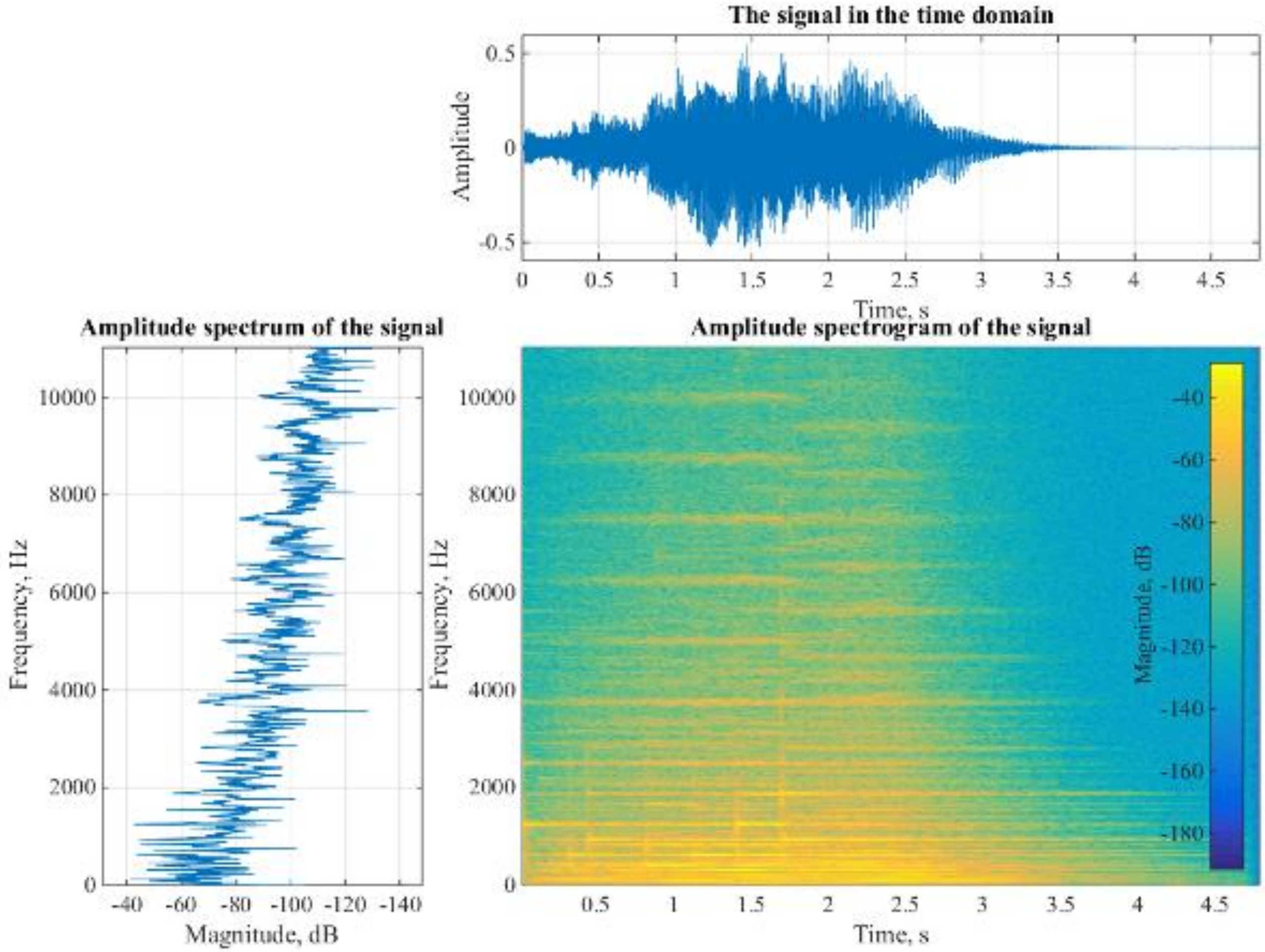
Self-Driving Cars	Speech-to-text	Fraud Detection	Customer Retention
Game Playing	Machine translation	Pricing	Proxy Models
Reducing Electricity Costs	Chatbots	Credit Risk	Call-Centre Routing
Analysing Satellite Photos	Recommender Systems	Sales Forecasting	Sentiment Analysis
Reading X-rays	Text-to-Speech	Anti-Money Laundering	Geographic Analysis

Classification

Regression and Classification

Self-Driving Cars	Speech-to-text	Fraud Detection	Customer Retention
Game Playing	Machine translation	Pricing	Proxy Models
Reducing Electricity Costs	Chatbots	Credit Risk	Call-Centre Routing
Analysing Satellite Photos	Recommender Systems	Sales Forecasting	Sentiment Analysis
Reading X-rays	Text-to-Speech	Anti-Money Laundering	Geographic Analysis

Digital Audio Files



- Digital Audio files are stored as a time series of arrays
- Each array contains information on pitch and loudness

Source: ch.mathworks.com

General Examples

- Self-Driving Cars
- Speech-to-text
- Fraud Detection
- Customer Retention
- Game Playing
- Machine translation
- Pricing
- Proxy Models
- Reducing Electricity Costs
- Chatbots
- Credit Risk
- Call-Centre Routing
- Analysing Satellite Photos
- Recommender Systems
- Sales Forecasting
- Sentiment Analysis
- Reading X-rays
- Text-to-Speech
- Anti-Money Laundering
- Geographic Analysis

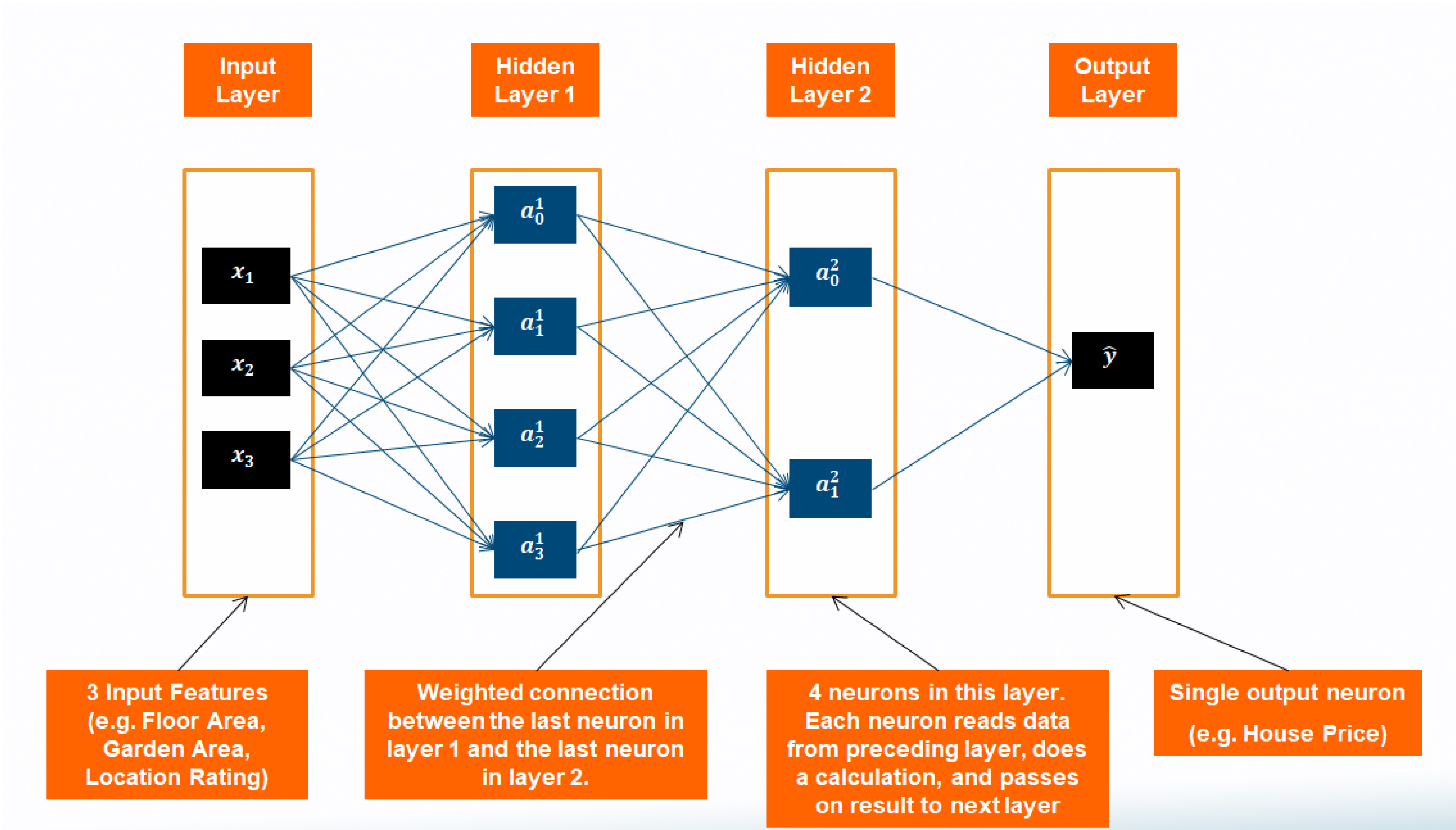
Agenda

- What is AI?
- Regression/Classification vs Specification
- How do Neural Networks work?
- Gradient Descent Optimisation
- Why Should Actuaries be Interested?
- Examples

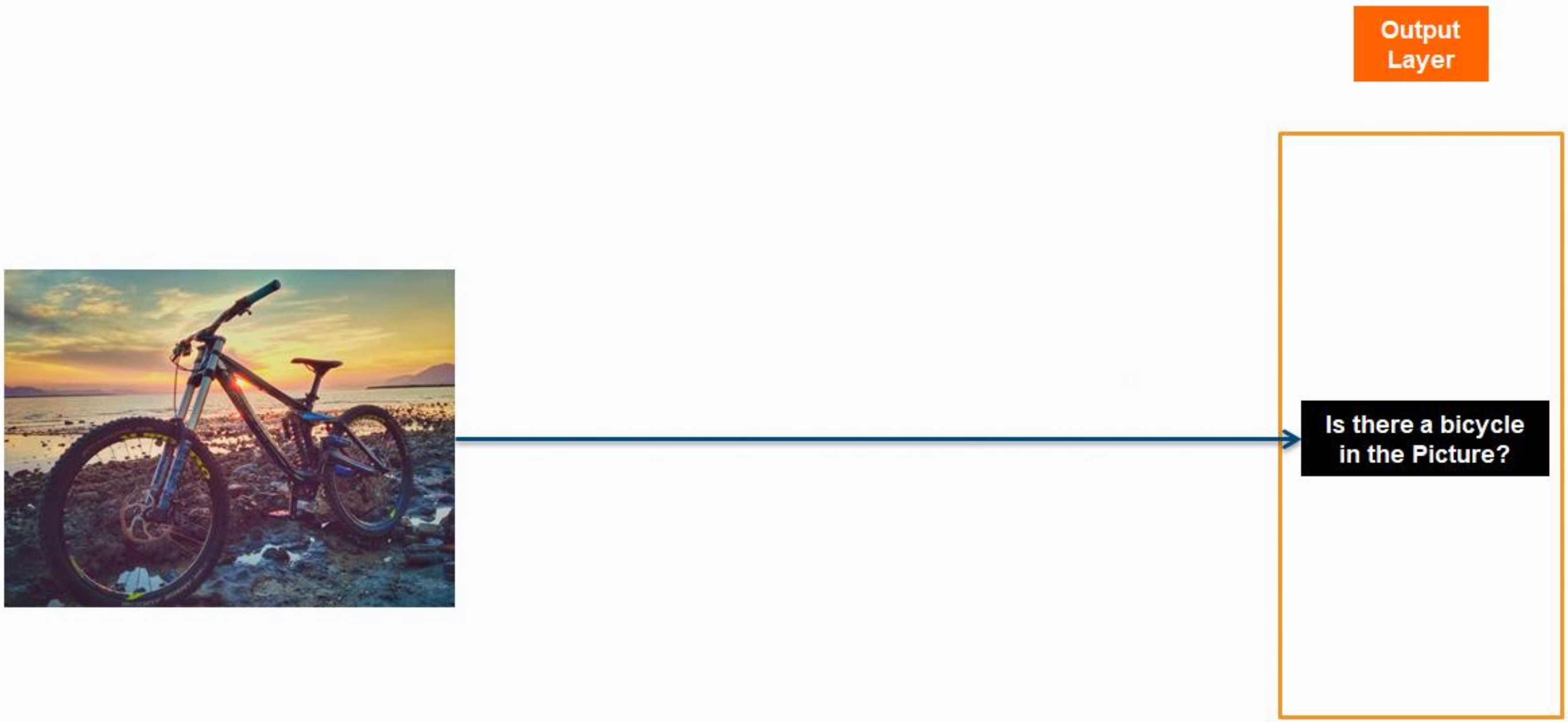
Universal Approximation Theorem

- In theory, neural networks can approximate *any* continuous function
- Corollory: Any task which can be approximated by a continuous function can be approximated by a neural network
  - Any task which can be specified using a continuous function can be approximated by a neural network

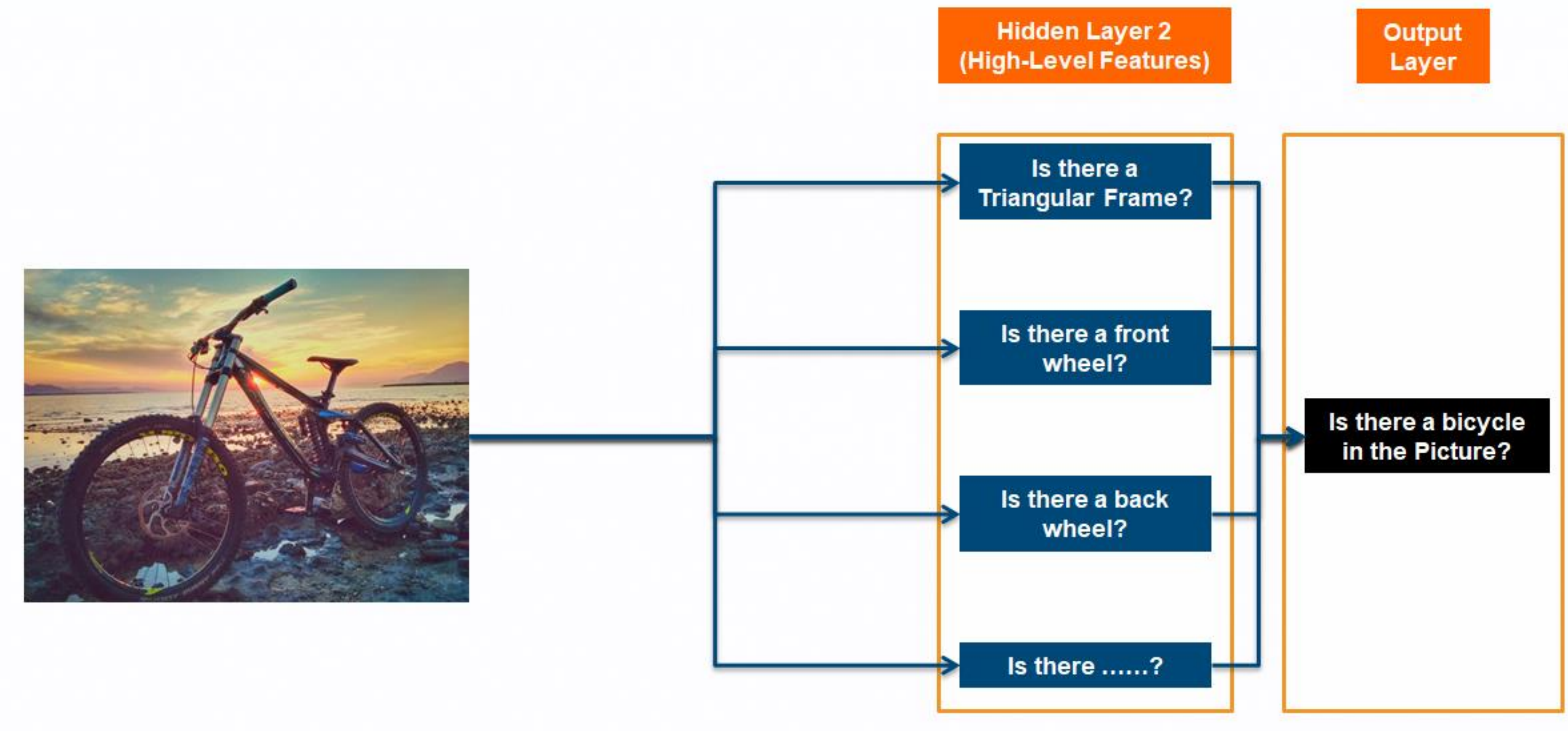
How do Neural Networks Work?



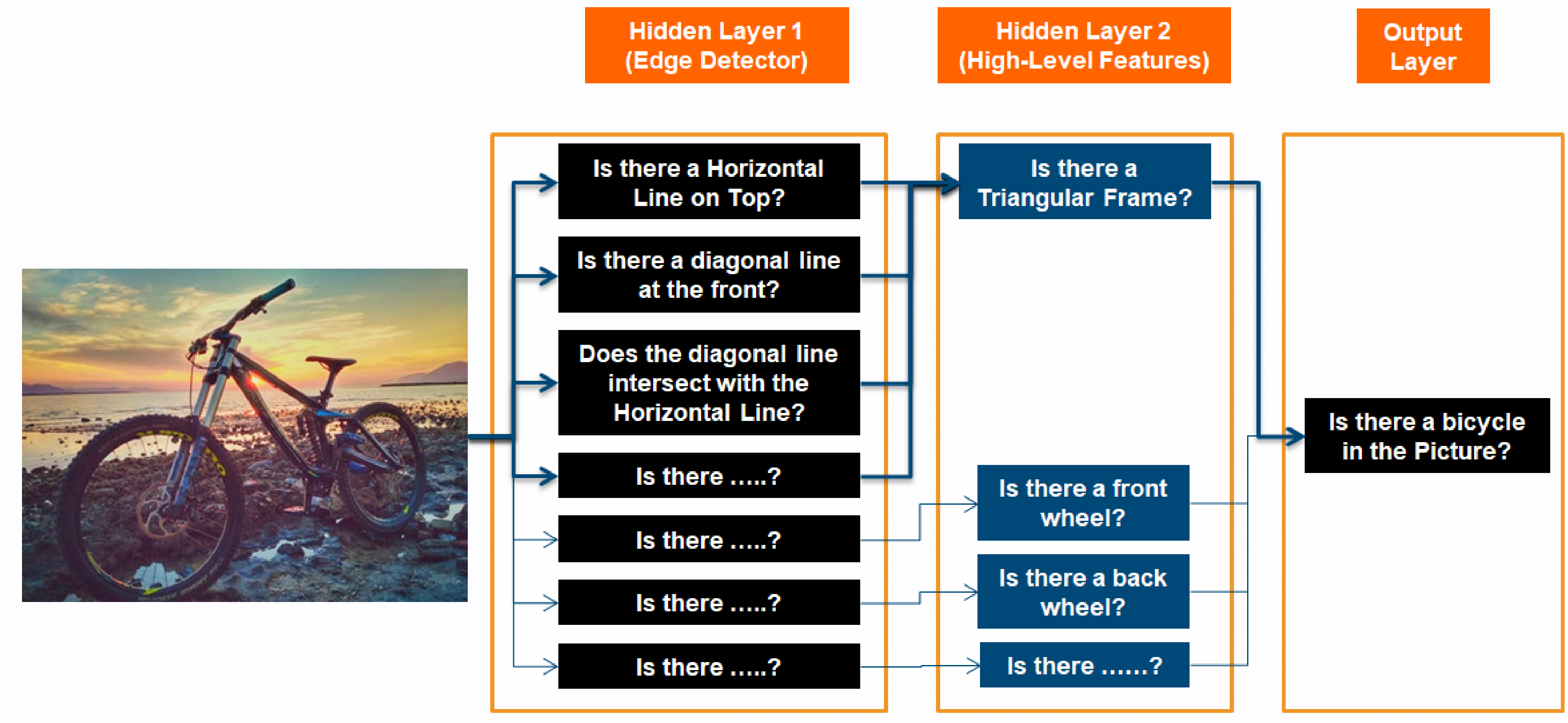
How do Neural Networks Work?



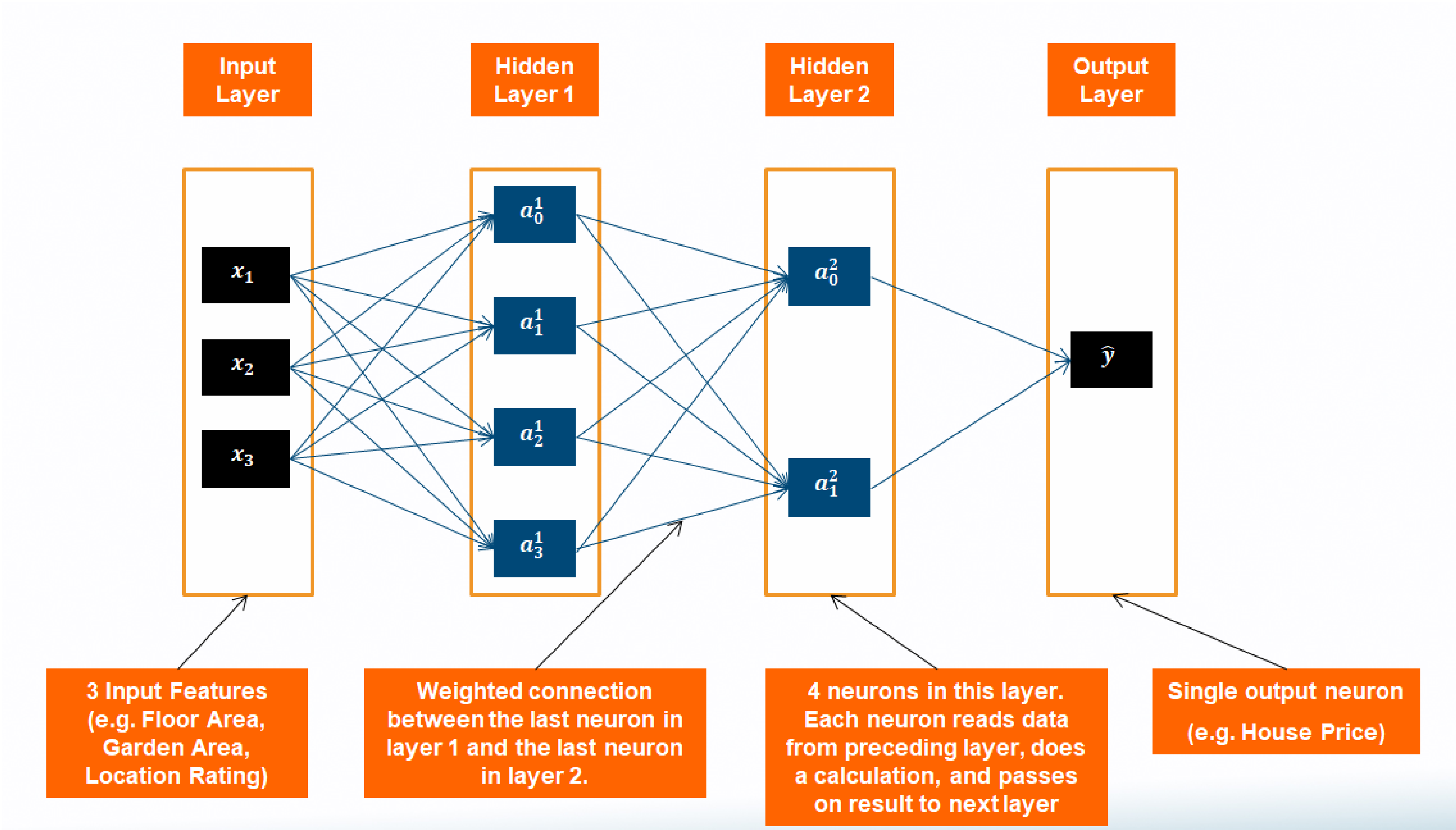
How do Neural Networks Work?



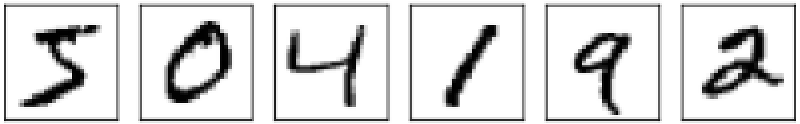
How do Neural Networks Work?



How do Neural Networks Work?

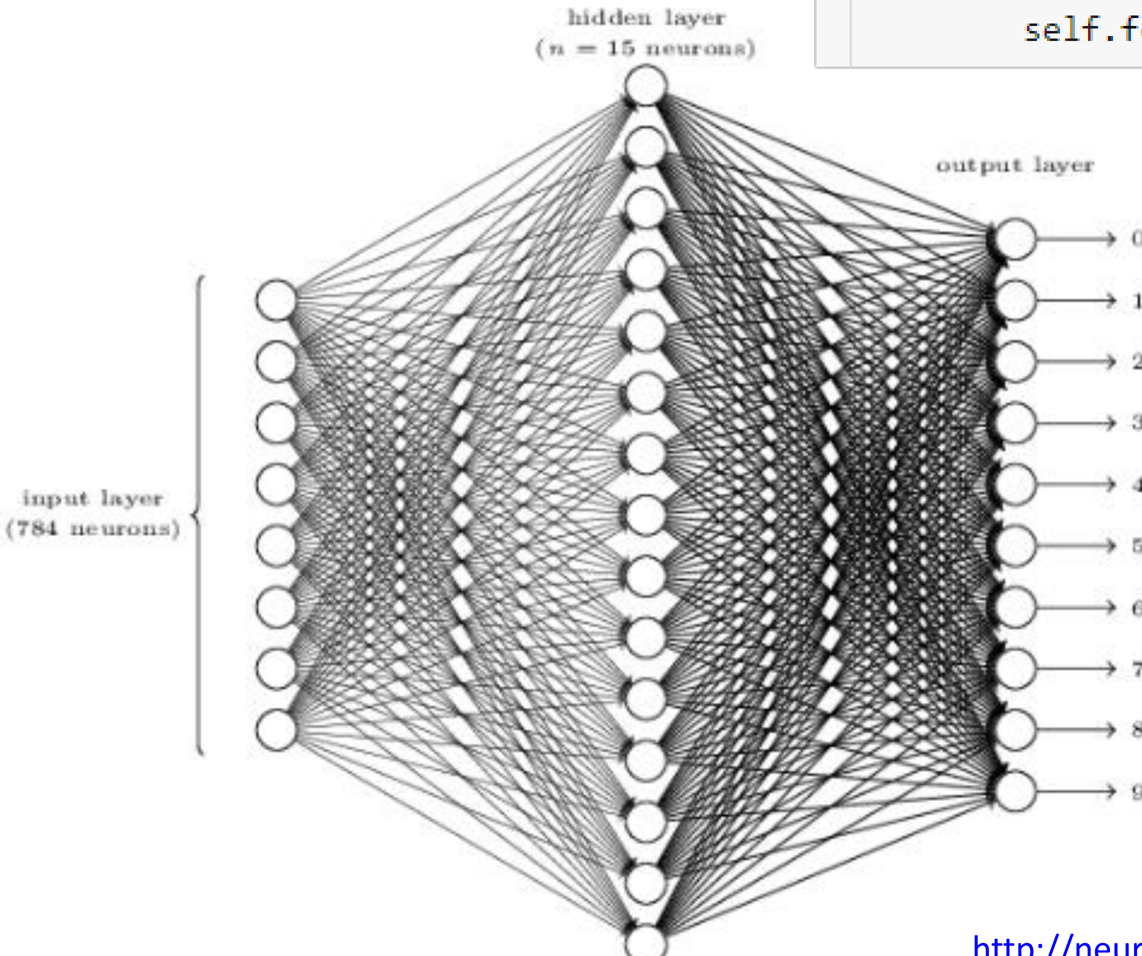


How do Neural Networks Work?



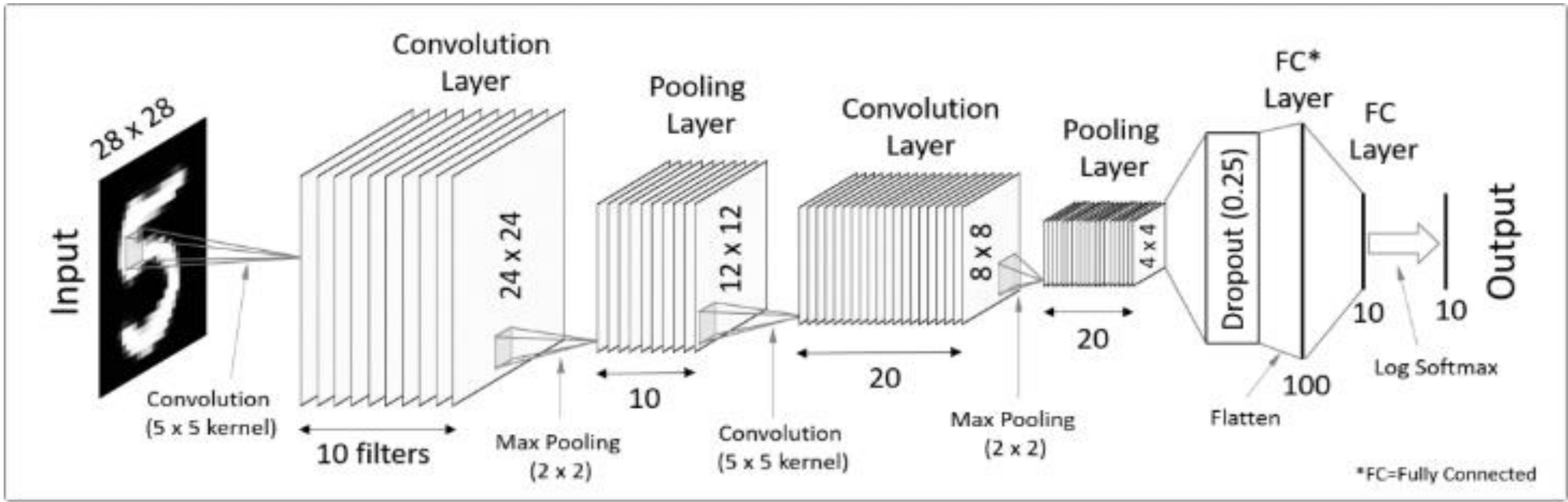

```python
import torch.nn as nn

class Net(nn.Module):
    def __init__(self):
        super(Net, self).__init__()
        self.fc1 = nn.Linear(784, 15)
        self.fc2 = nn.Linear(15,  10)
```



http://neuralnetworksanddeeplearning.com

How do Neural Networks Work?



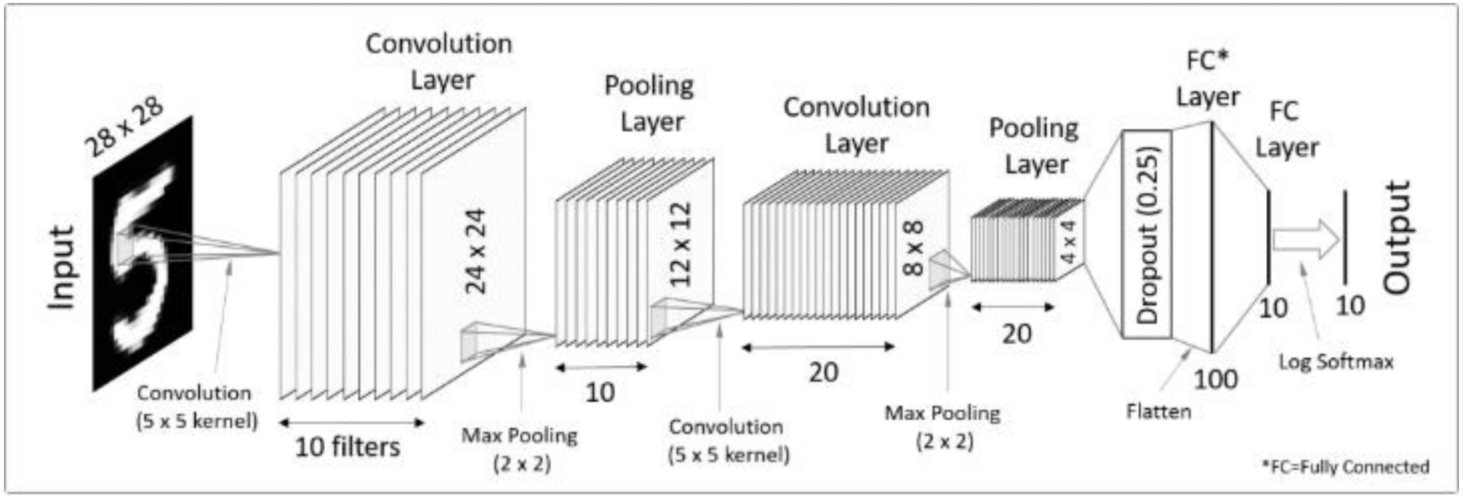
https://codetolight.wordpress.com/2017/11/29/getting-started-with-pytorch-for-deep-learning-part-3-neural-network-basics/

How do Neural Networks Work?

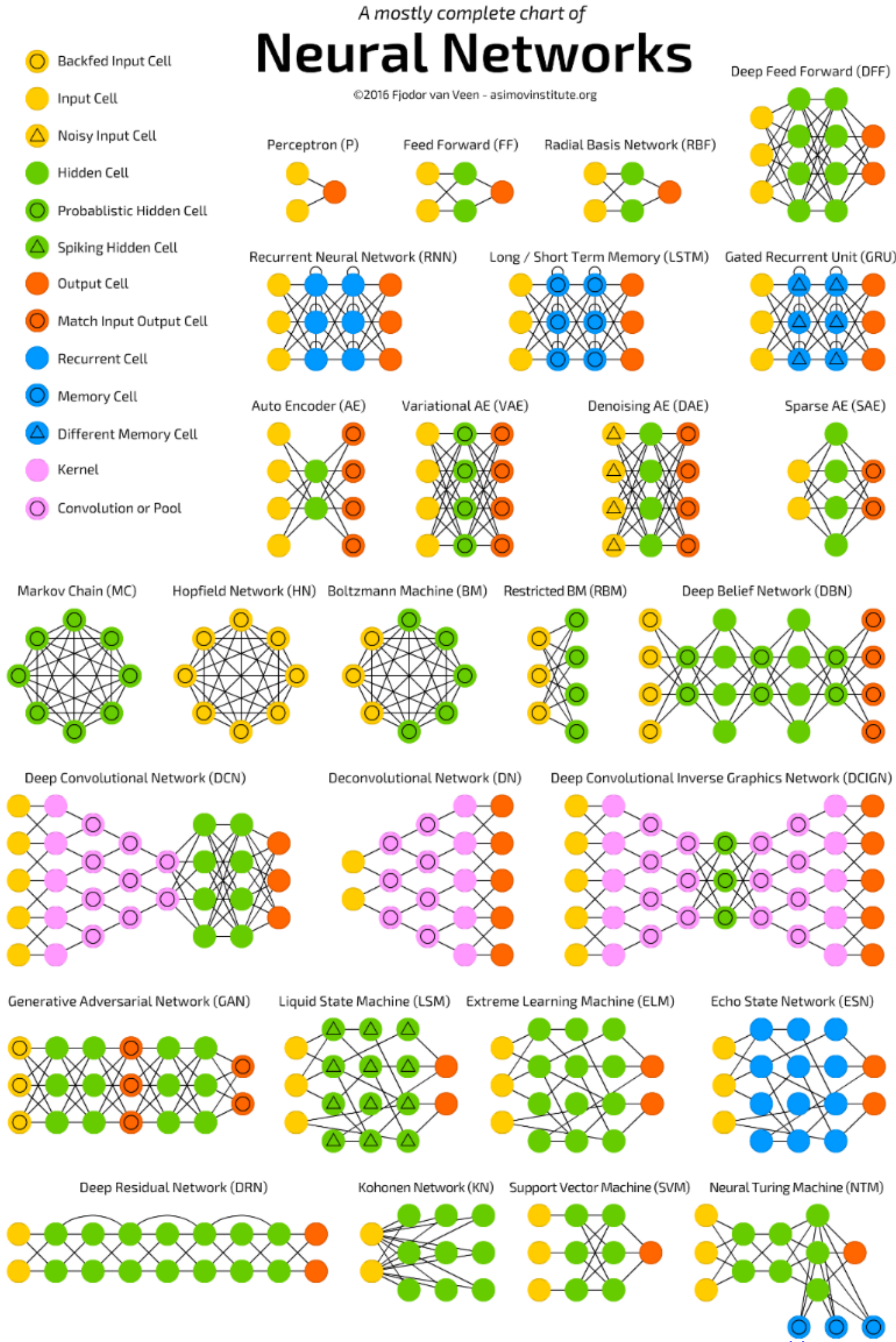
```
class Net(nn.Module):
    def __init__(self):
        super(Net, self).__init__()
        # define all the components that will be used in the NN
        self.conv1 = nn.Conv2d(1, 10, kernel_size=5)
        self.conv2 = nn.Conv2d(10, 20, kernel_size=5)
        self.mp = nn.MaxPool2d(2, padding=0)
        self.drop2D = nn.Dropout2d(p=0.25, inplace=False)
        self.fc1 = nn.Linear(320,100)
        self.fc2 = nn.Linear(100,10)
```

https://codetolight.wordpress.com/2017/11/29/getting-started-with-pytorch-for-deep-learning-part-3-neural-network-basics/

How do Neural Networks Work?



https://codetolight.wordpress.com/2017/11/29/getting-started-with-pytorch-for-deep-learning-part-3-neural-network-basics/



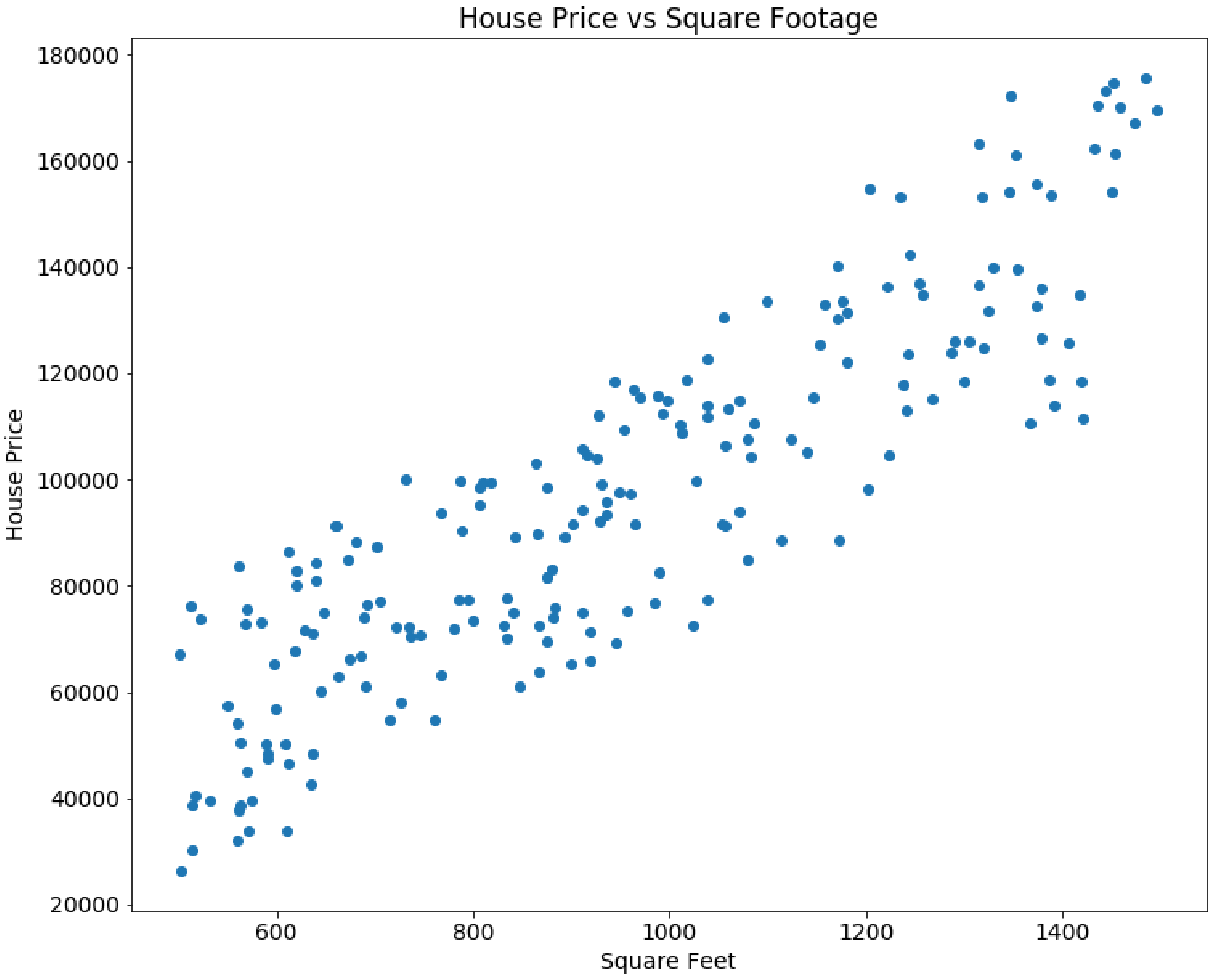
http://www.asimovinstitute.org/neural-network-zoo/

Agenda

- What is AI?
- Regression/Classification vs Specification
- How do Neural Networks work?
- Gradient Descent Optimisation
- Why Should Actuaries be Interested?
- Examples

Practical Example: Traditional Modelling and Machine Learning

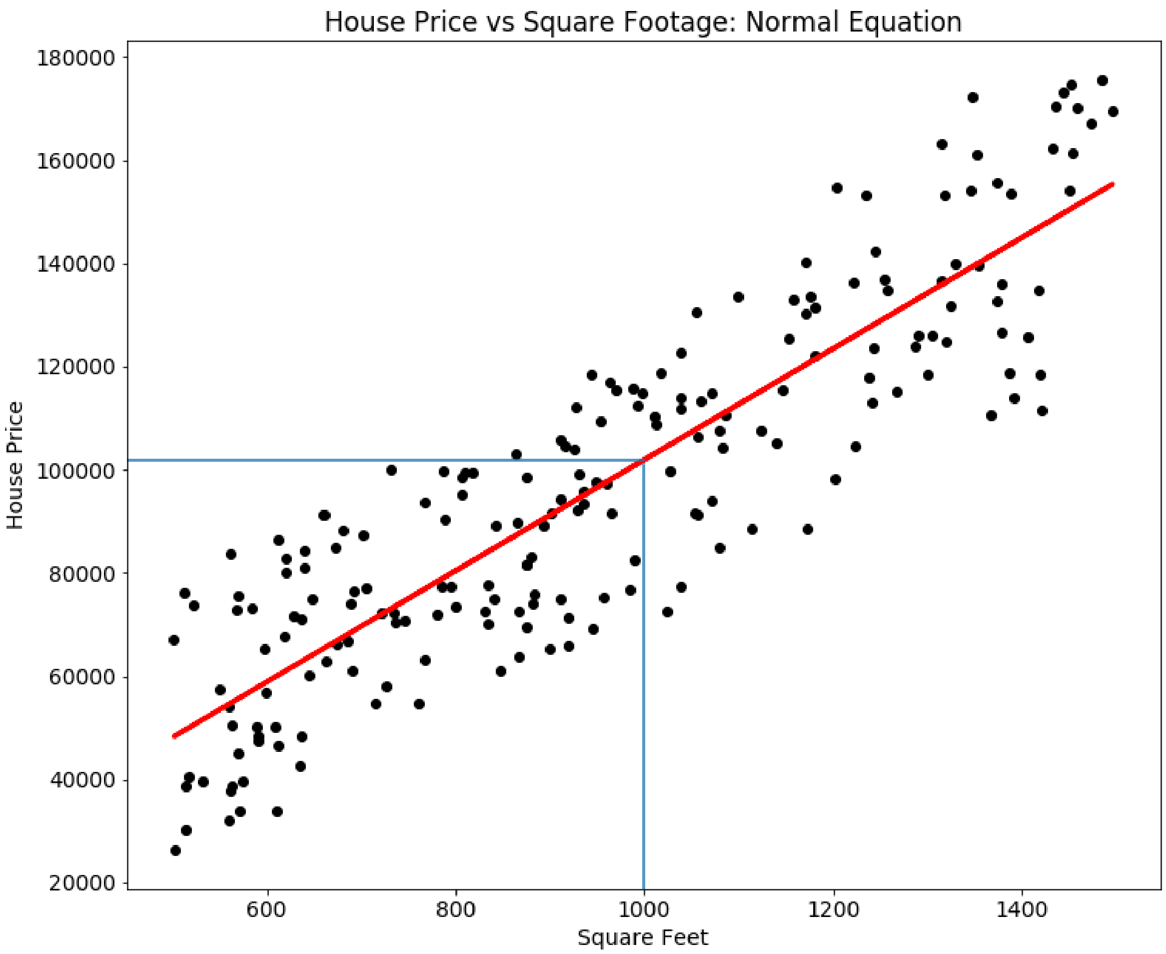
How much is a 1000 square foot house?



Eyeball approach:

Around €90k

Linear Regression Predictive Model



- Linear Regression Model:
  - Price = €101,955
  - Slope = 108
  - Intercept = -5,700
  - MSE = 258 million
- But how do you find the slope and intercept?

Functional Specification Approach: Normal Equation

Linear Regression Model:

$$\hat{y} = ax + b = \theta X$$

where:

- $\theta = [a \;\; b]$
- $X = [x \;\; 1]$

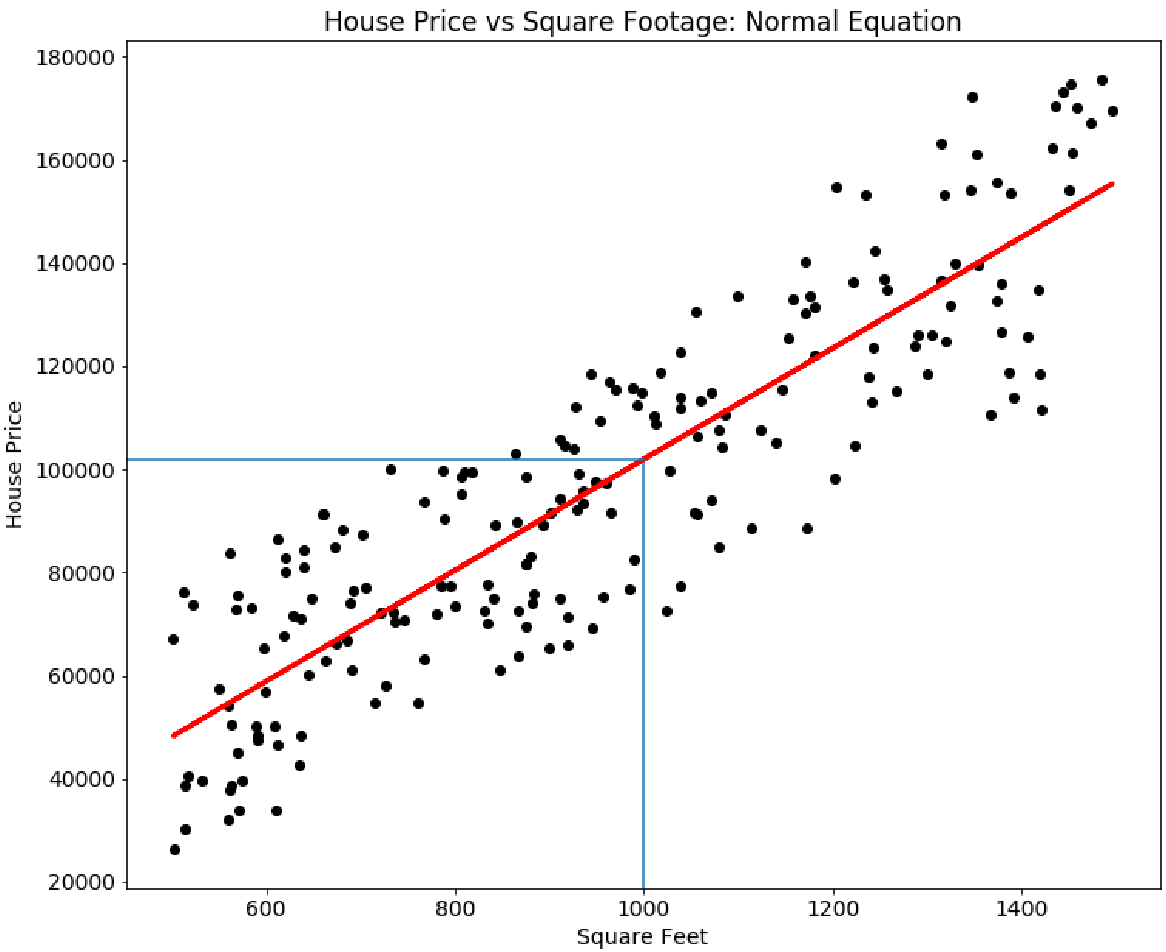
Choose Loss Function, such as Mean Squared Error

Calculate parameters theta using formula:

$$\theta = (X^T X)^{-1} X^T y$$

```
theta = (np.linalg.pinv(X.T * X) * X.T) * Y
y_hat = X * theta
```

Linear Regression Predictive Model



Linear Regression Model:

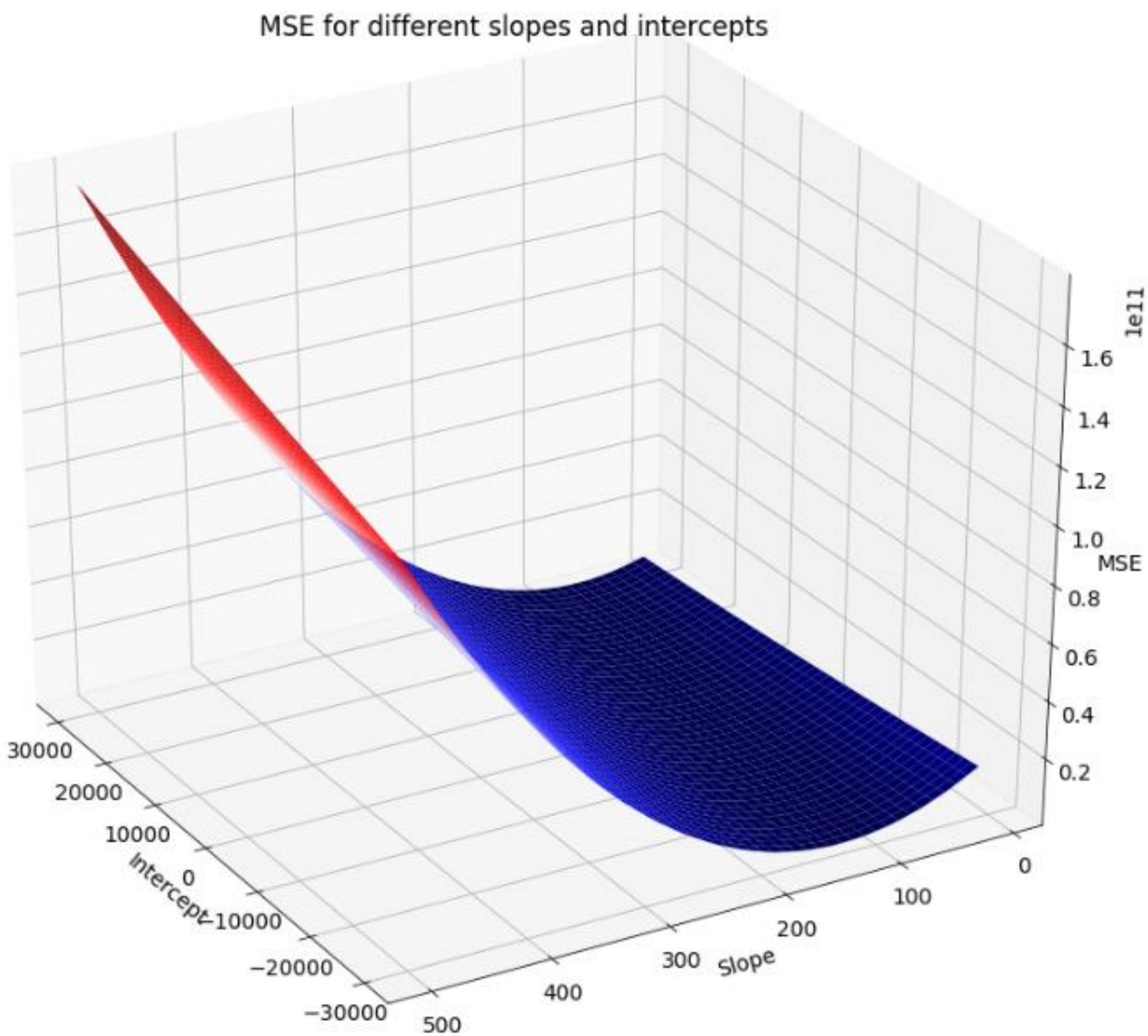
- Price = €101,955
- Slope = 108
- Intercept = -5,700
- MSE = 258 million

Approach 1: Normal Equation

Problem with normal equation:

- Only works if $X^T X$ is invertible
- Doesn’t work on other models
- Doesn’t work well on large datasets

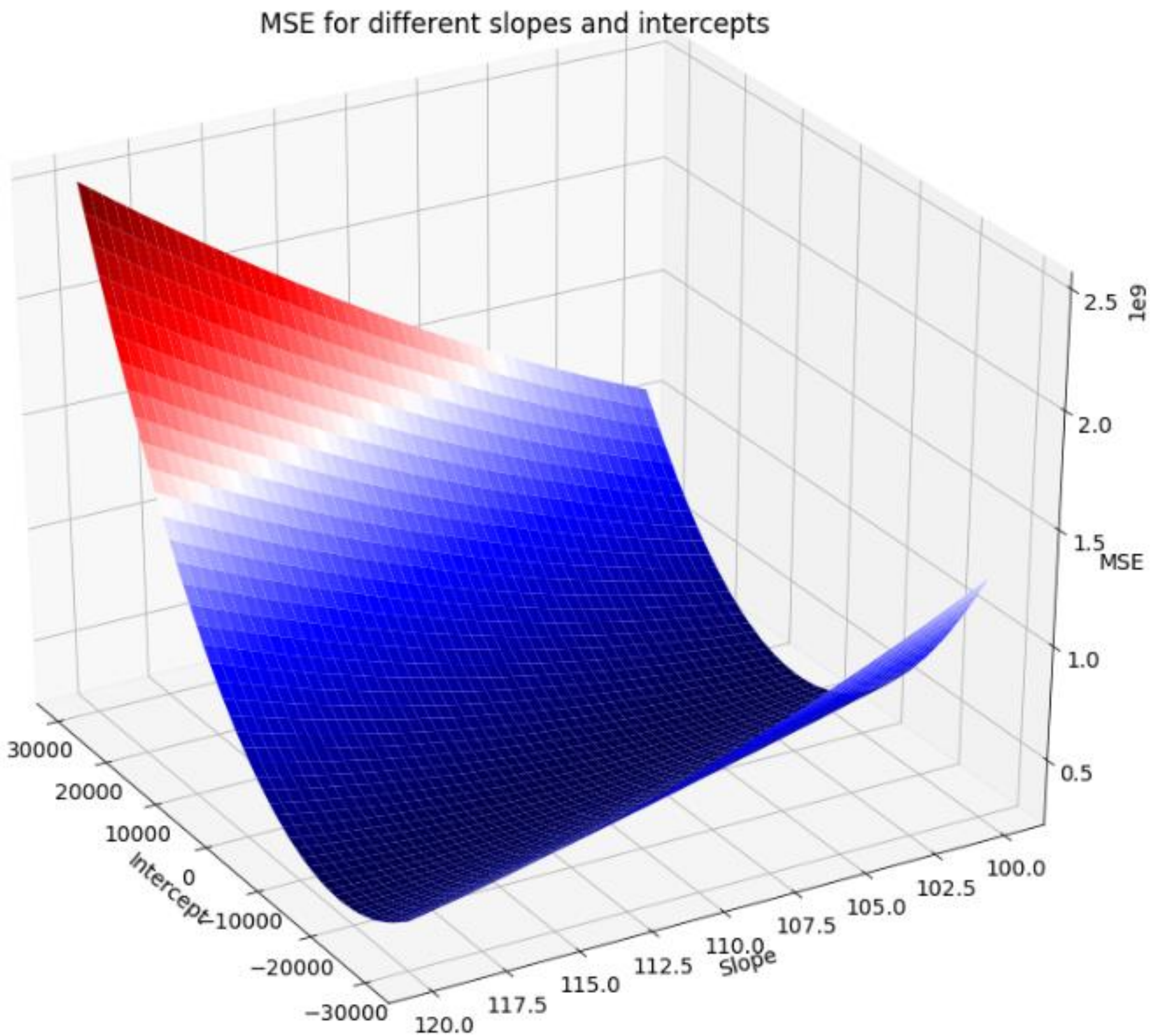
Approach 2: Gridsearch




```
Point with minimum MSE:
```

	86
Slopes	113.16
Intercepts	-11,052.63
MSE	261,059,459.22

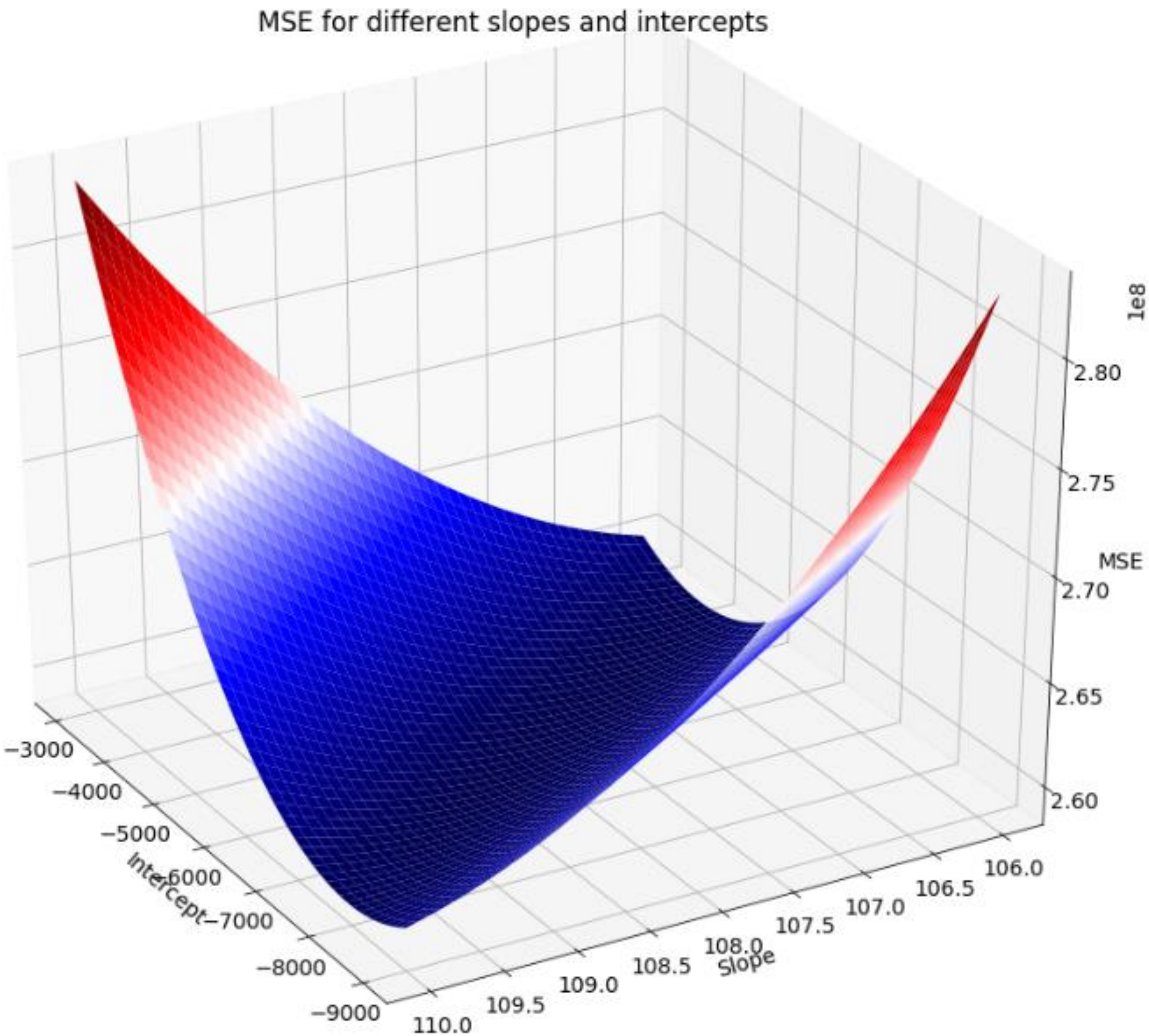
Approach 2: Gridsearch




```
Point with minimum MSE:
```

	128
Slopes	106.32
Intercepts	-4,736.84
MSE	258,939,860.54

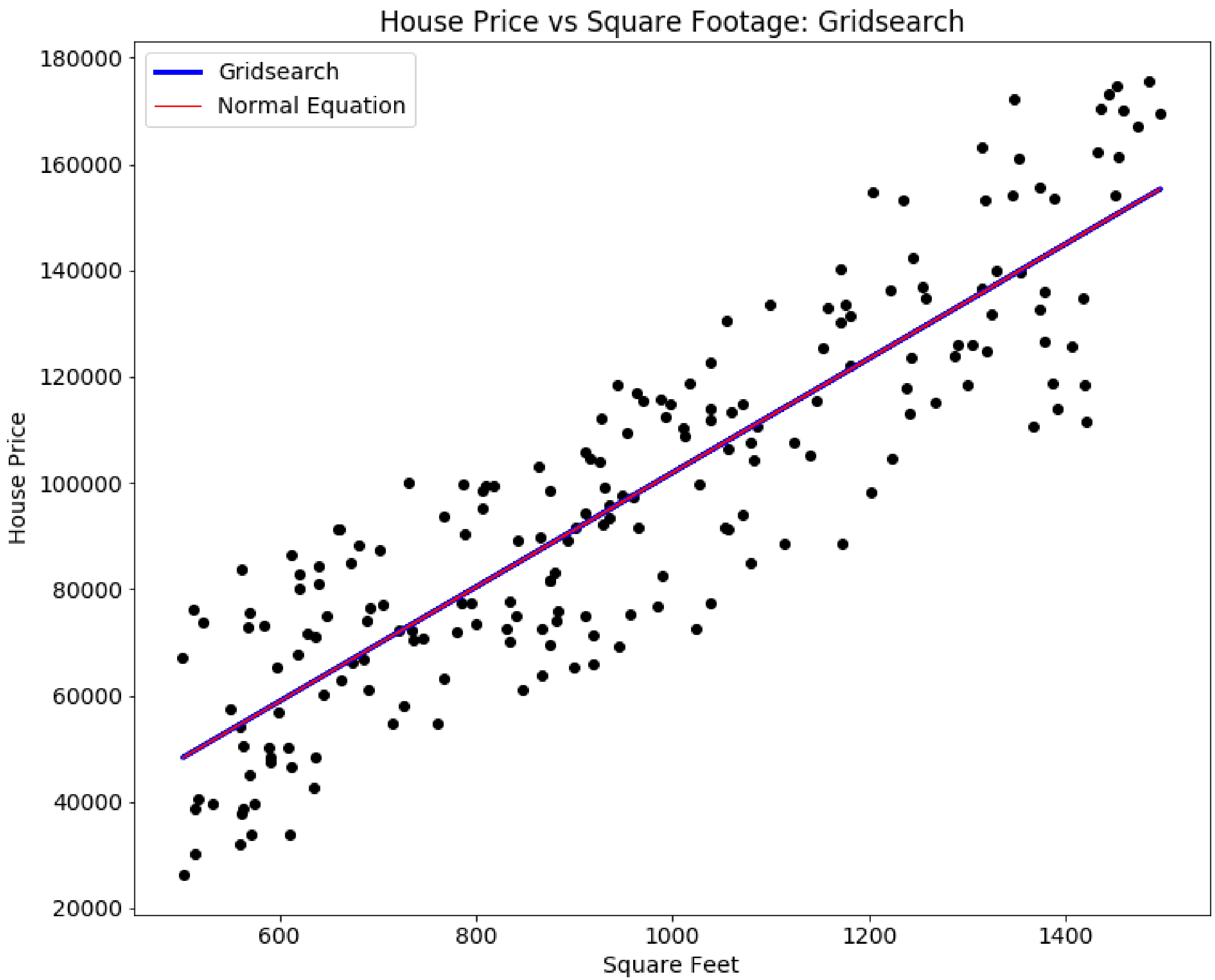
Approach 2: Gridsearch




```
Point with minimum MSE:
```

	1708
Slopes	107.68
Intercepts	-5,743.72
MSE	258,689,013.27

Approach 2: Gridsearch



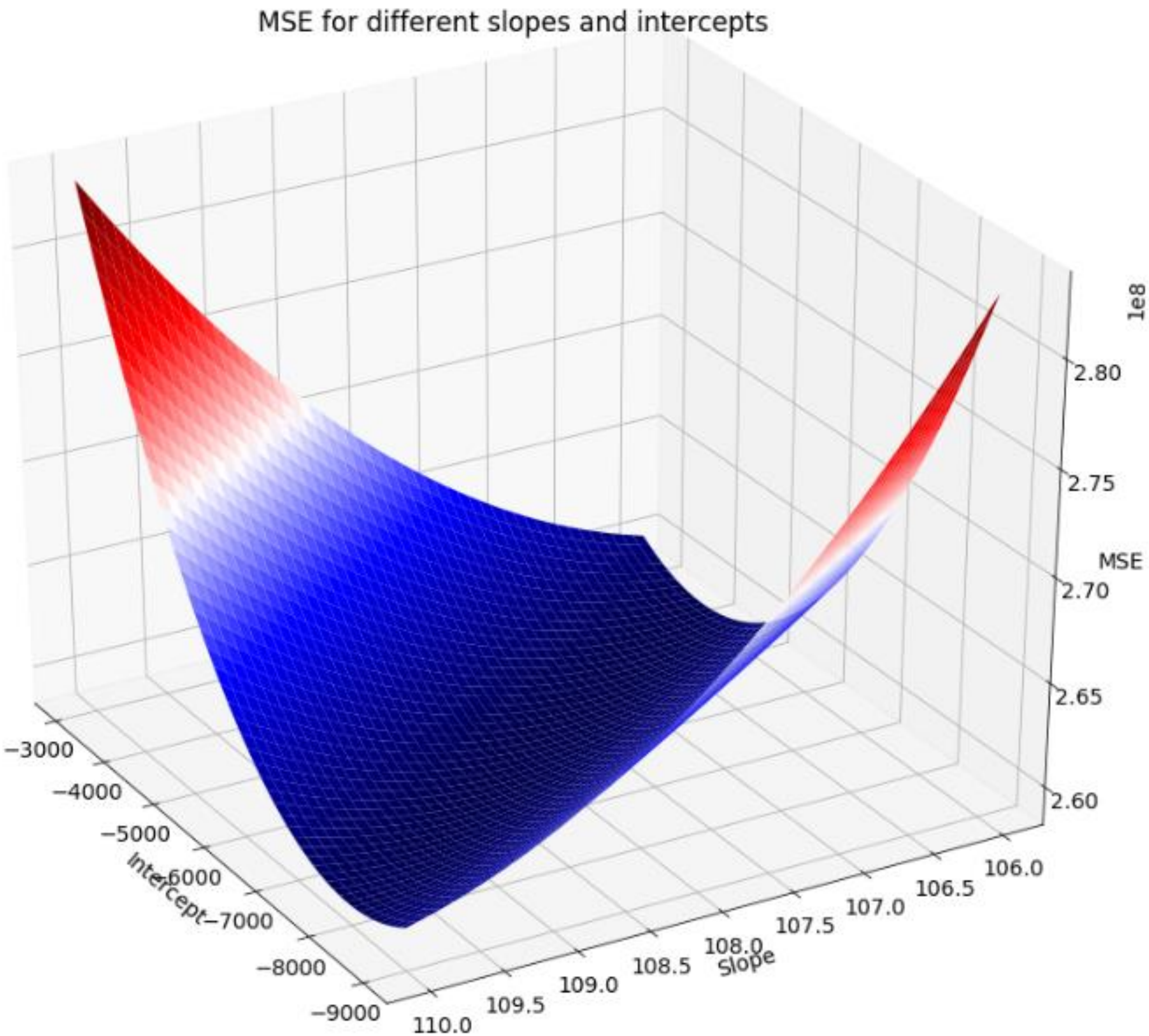
Point with minimum MSE:

	1708
Slopes	107.68
Intercepts	-5,743.72
MSE	258,689,013.27

Approach 2: Gridsearch

- Problem with gridsearch: Very inefficient
  - Only works for models with a handful of parameters

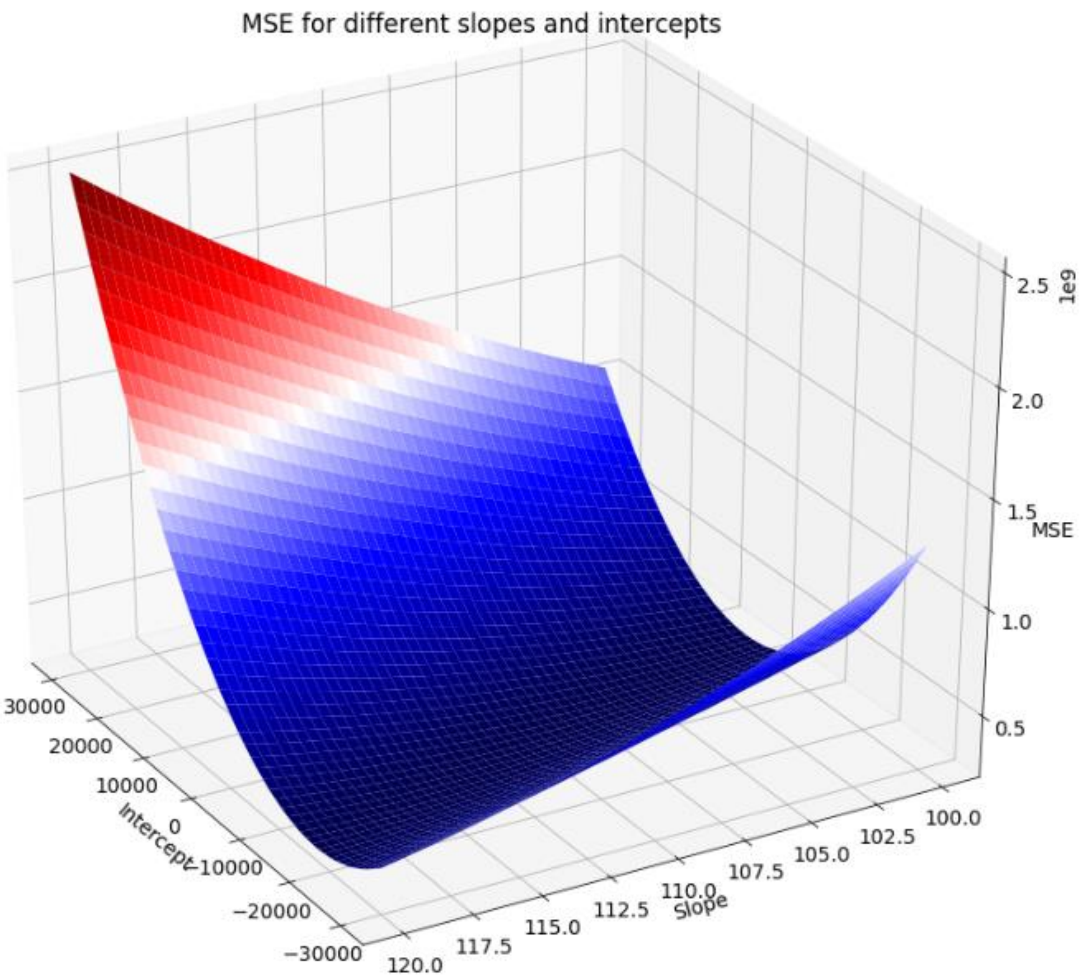
Approach 2: Gridsearch




```
Point with minimum MSE:
```

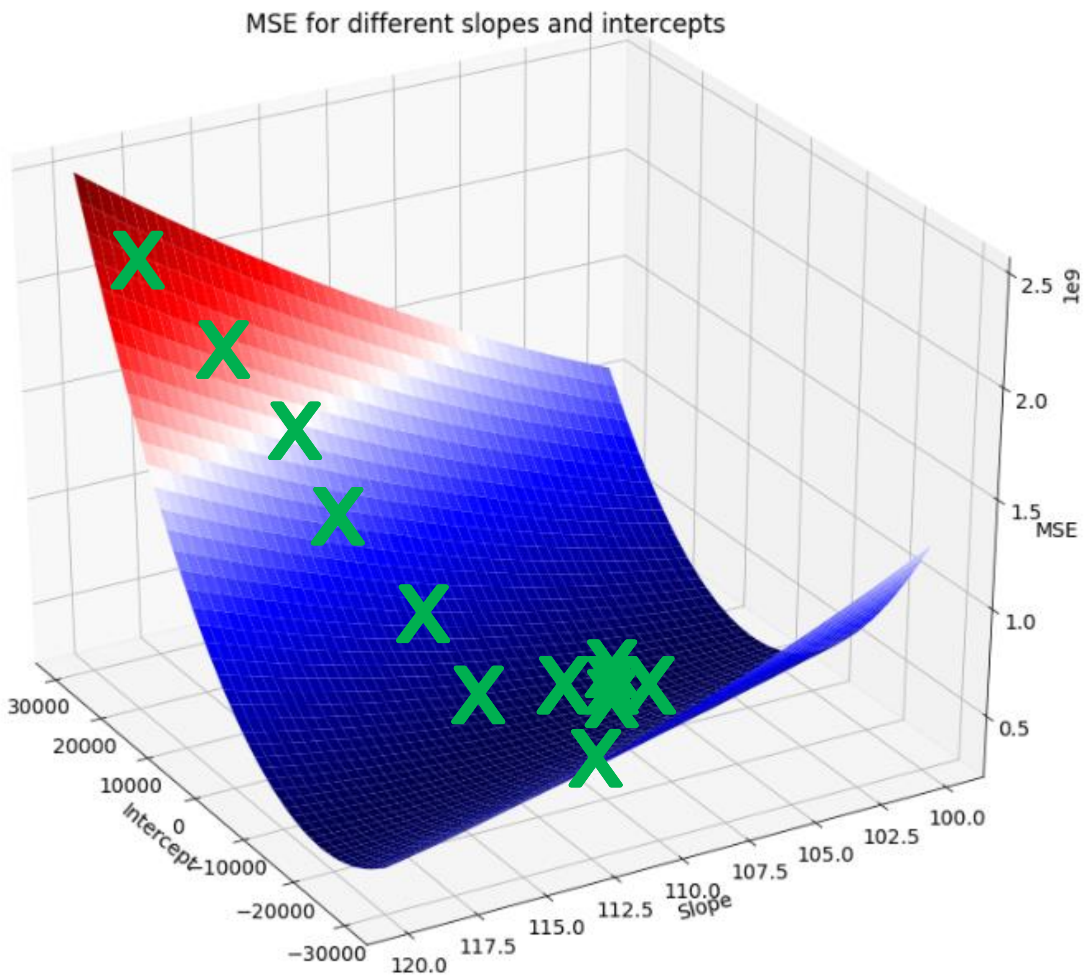
	1708
Slopes	107.68
Intercepts	-5,743.72
MSE	258,689,013.27

Approach 3: Stochastic Gradient Descent



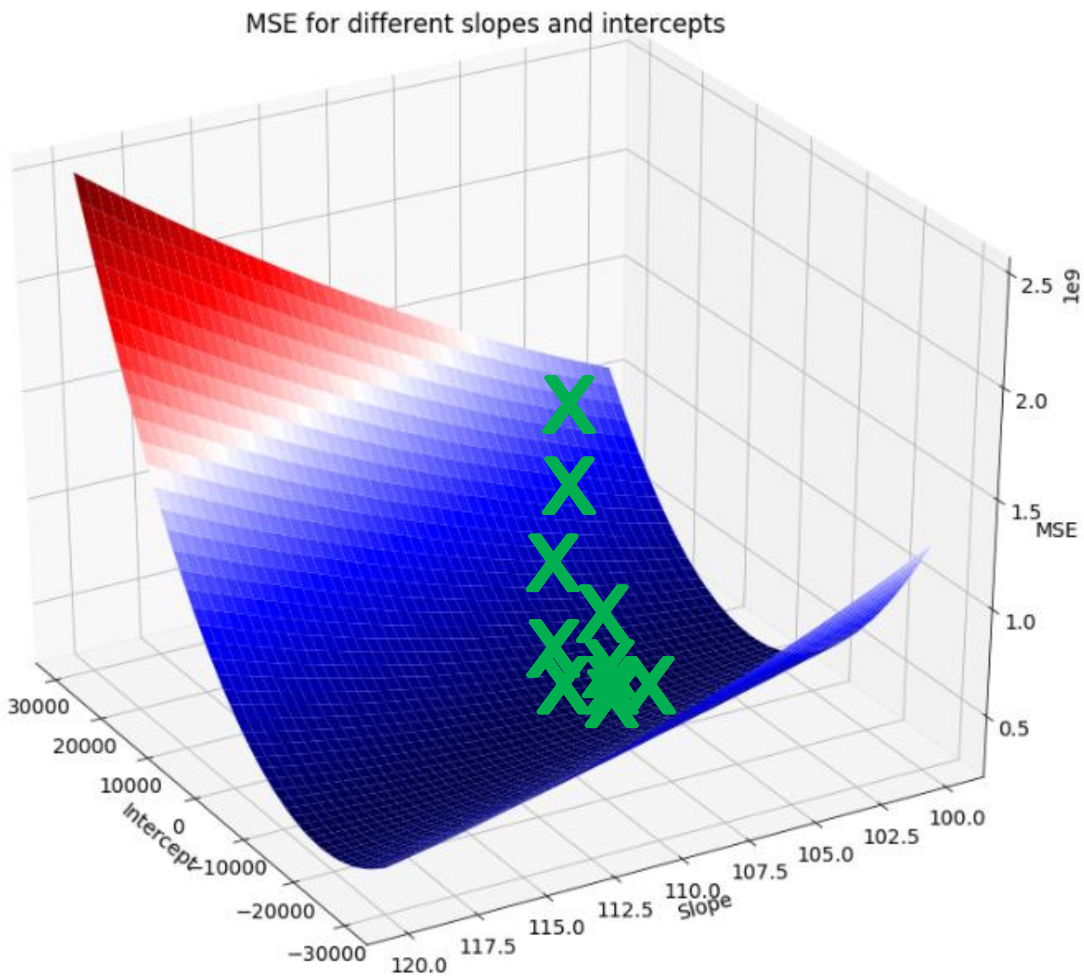
1. You don’t know the slope and intercept, so randomly choose them
2. Therefore you start at a random point
3. Calculate the slope of the MSE loss surface at that point
4. Take a step downhill
5. Repeat 3 and 4 until you reach the lowest point on the loss surface

Approach 3: Stochastic Gradient Descent



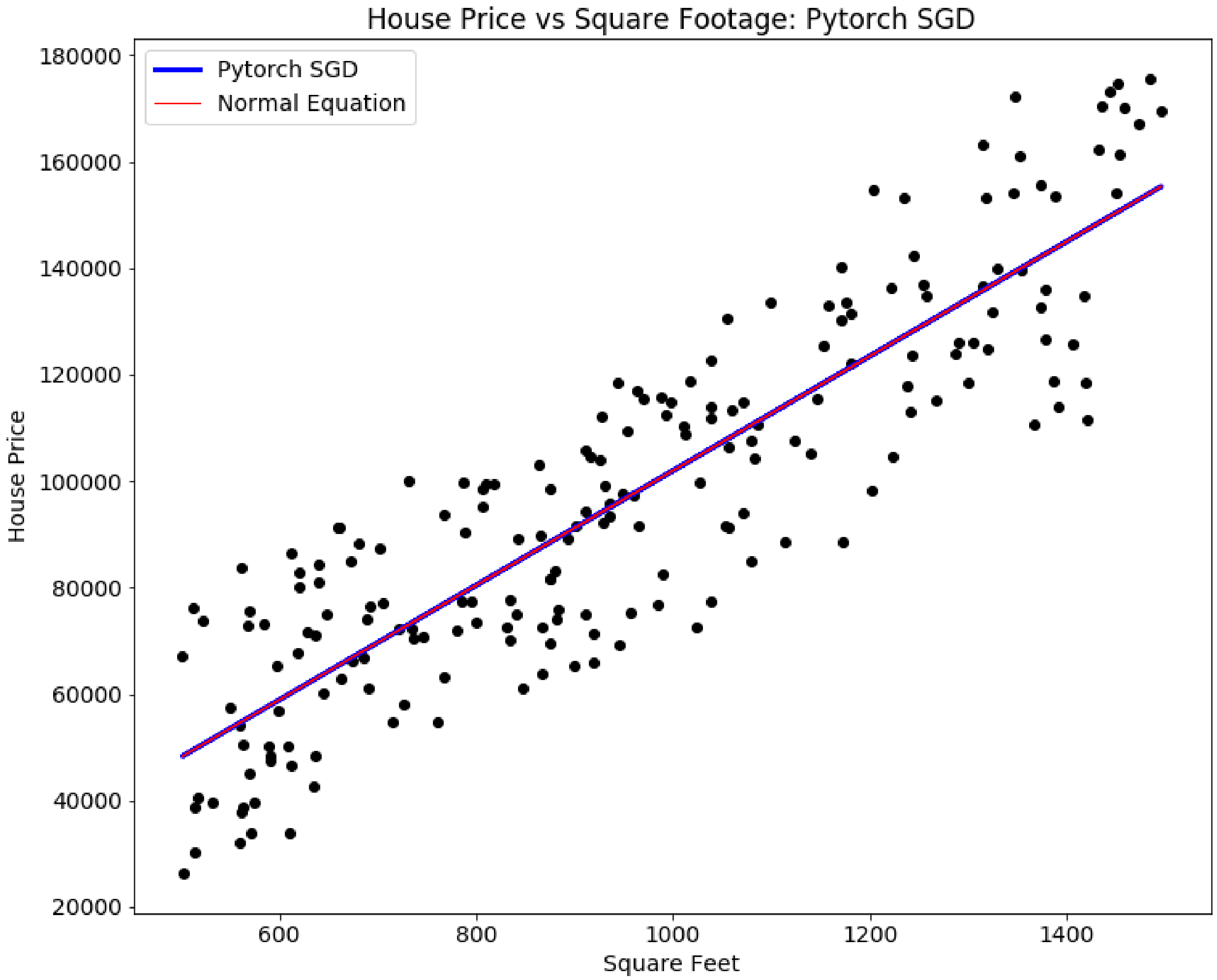
1. You don't know the slope and intercept, so randomly choose them
2. Therefore you start at a random point
3. Calculate the slope of the MSE loss surface at that point
4. Take a step downhill
5. Repeat 3 and 4 until you reach the lowest point on the loss surface

Approach 3: Stochastic Gradient Descent



1. You don't know the slope and intercept, so randomly choose them
2. Therefore you start at a random point
3. Calculate the slope of the MSE loss surface at that point
4. Take a step downhill
5. Repeat 3 and 4 until you reach the lowest point on the loss surface

Approach 3: Stochastic Gradient Descent



SGD gives exact same answer as Normal Equation in this example

SGD: Python Code

```
a = Variable(torch.ones(1,1), requires_grad=True)
b = Variable(torch.ones(1,1), requires_grad=True)

optimizer = torch.optim.SGD([a, b],lr=0.0001) # Use SGD machine-learning algorithm
loss_fn   = torch.nn.MSELoss()                # Use Mean-Squared-Error loss metric

for i in range(50000):                        # Take 50k steps downhill
    y_hat = a*x + b                           # Model
    loss = loss_fn(y_hat, y)                  # Calculate MSE for this particular model
    optimizer.zero_grad()
    loss.backward()                           # Calculate slope of loss surface
    optimizer.step()                          # Step downhill
```

Approach 3: Stochastic Gradient Descent

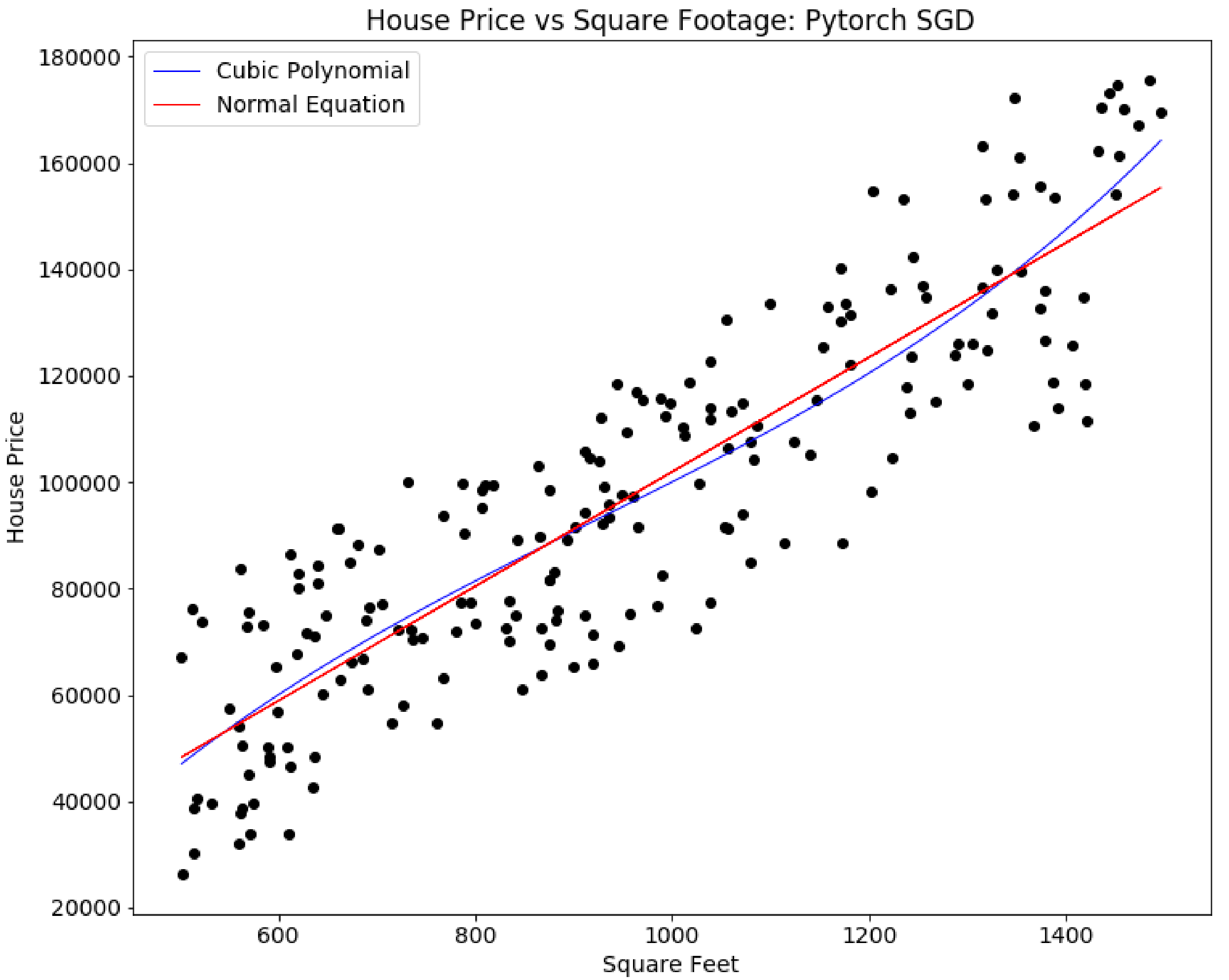
SGD: Cubic Polynomial

```
a = Variable(torch.ones(1,1), requires_grad=True)
b = Variable(torch.ones(1,1), requires_grad=True)
c = Variable(torch.ones(1,1), requires_grad=True)
d = Variable(torch.ones(1,1), requires_grad=True)

optimizer = torch.optim.SGD([a, b, c, d],lr=0.0001) # Use SGD machine-learning algorithm
loss_fn = torch.nn.MSELoss()                        # Use Mean-Squared-Error loss metric

for i in range(100000):                             # Take 100k steps downhill
    y_hat_2 = a*x**3 + b*x**2 + c*x + d            # Model
    loss = loss_fn(y_hat_2, y)                      # Calculate MSE for this particular model
    optimizer.zero_grad()
    loss.backward()                                 # Calculate slope of MSE loss surface
    optimizer.step()                                # Step downhill
```

SGD: Cubic Polynomial



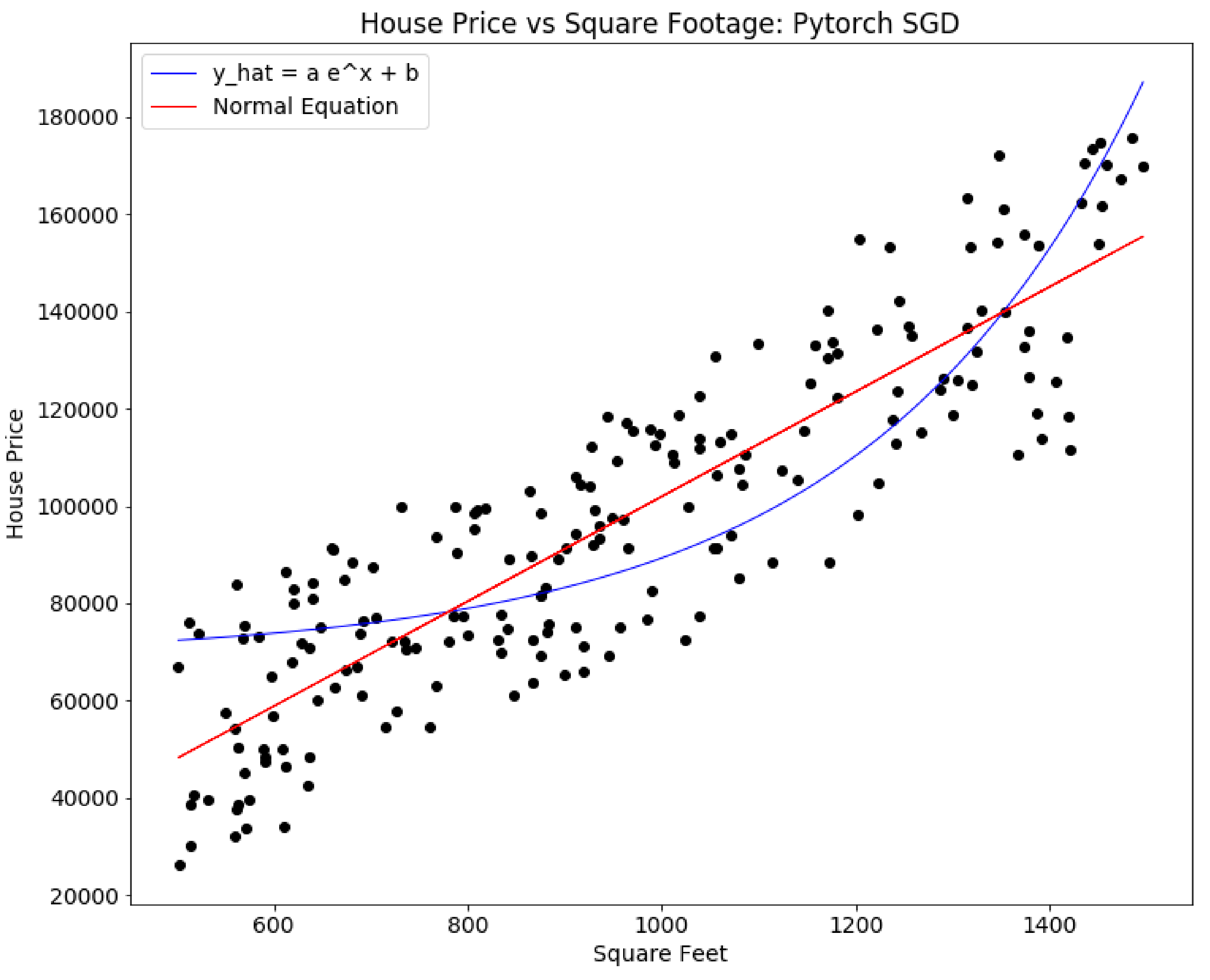
SGD: Exponential Model

```
a = Variable(torch.ones(1,1), requires_grad=True)
b = Variable(torch.ones(1,1), requires_grad=True)

optimizer = torch.optim.SGD([a, b],lr=0.0001) # Use SGD machine-learning algorithm
loss_fn = torch.nn.MSELoss()      # Use Mean-Squared-Error loss metric

for i in range(50000):            # Take 50k steps downhill
    y_hat_2 = a*np.exp(x) + b     # Model
    loss = loss_fn(y_hat_2, y)    # Calculate MSE for this particular model
    optimizer.zero_grad()
    loss.backward()               # Calculate slope of MSE loss surface
    optimizer.step()              # Step downhill
```

SGD: Exponential Curve



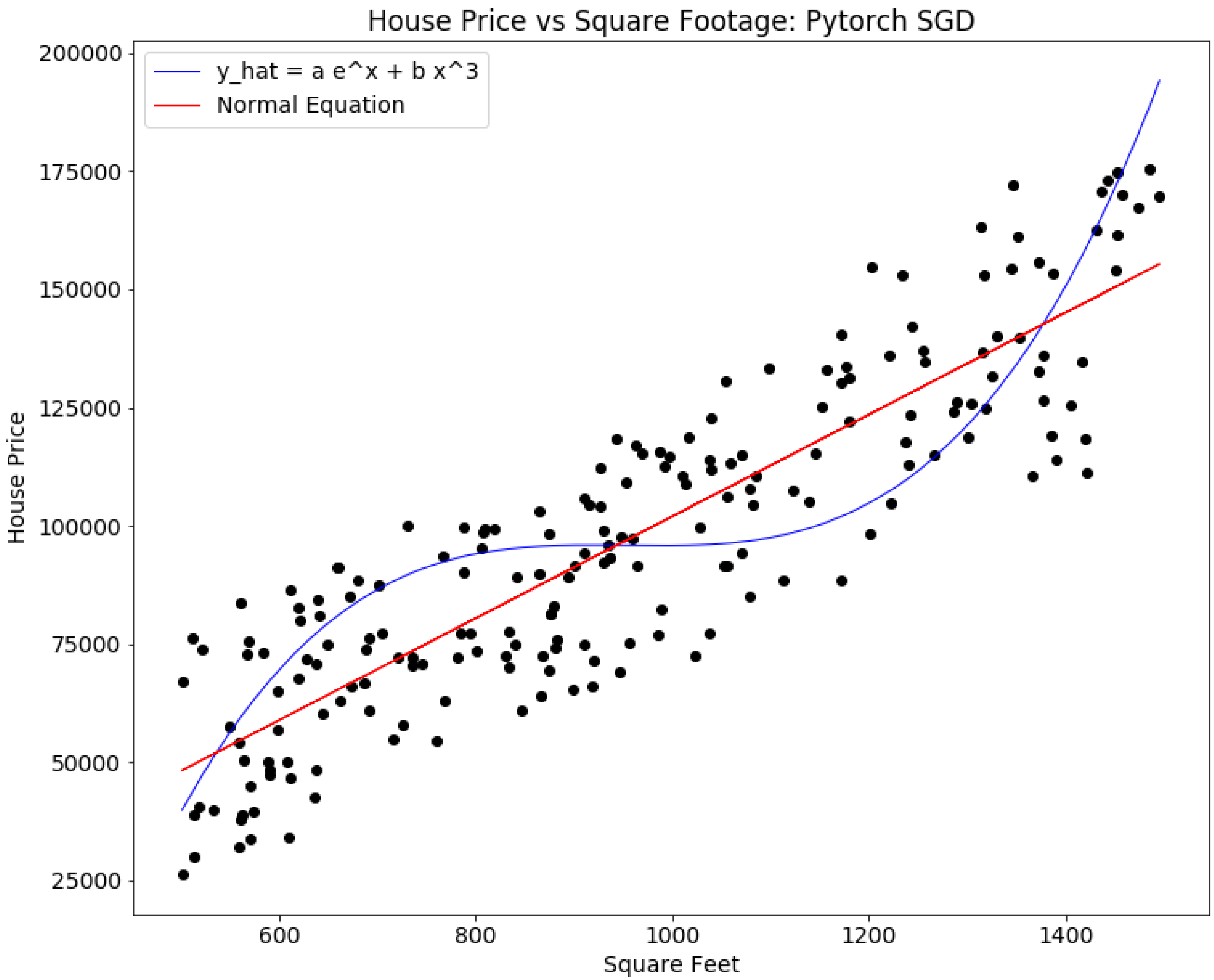
SGD: Exponential Plus Cubic Model

```python
a = Variable(torch.ones(1,1), requires_grad=True)
b = Variable(torch.ones(1,1), requires_grad=True)

optimizer = torch.optim.SGD([a, b],lr=0.0001) # Use SGD machine-learning algorithm
loss_fn = torch.nn.MSELoss()                  # Use Mean-Squared-Error loss metric

for i in range(100000):                       # Take 100k steps downhill
    y_hat_2 = a*np.exp(x) + b*x**3            # Model
    loss = loss_fn(y_hat_2, y)                # Calculate MSE for this particular model
    optimizer.zero_grad()
    loss.backward()                           # Calculate slope of MSE loss surface
    optimizer.step()                          # Step downhill
```

SGD: Exponential Plus Cubic Model



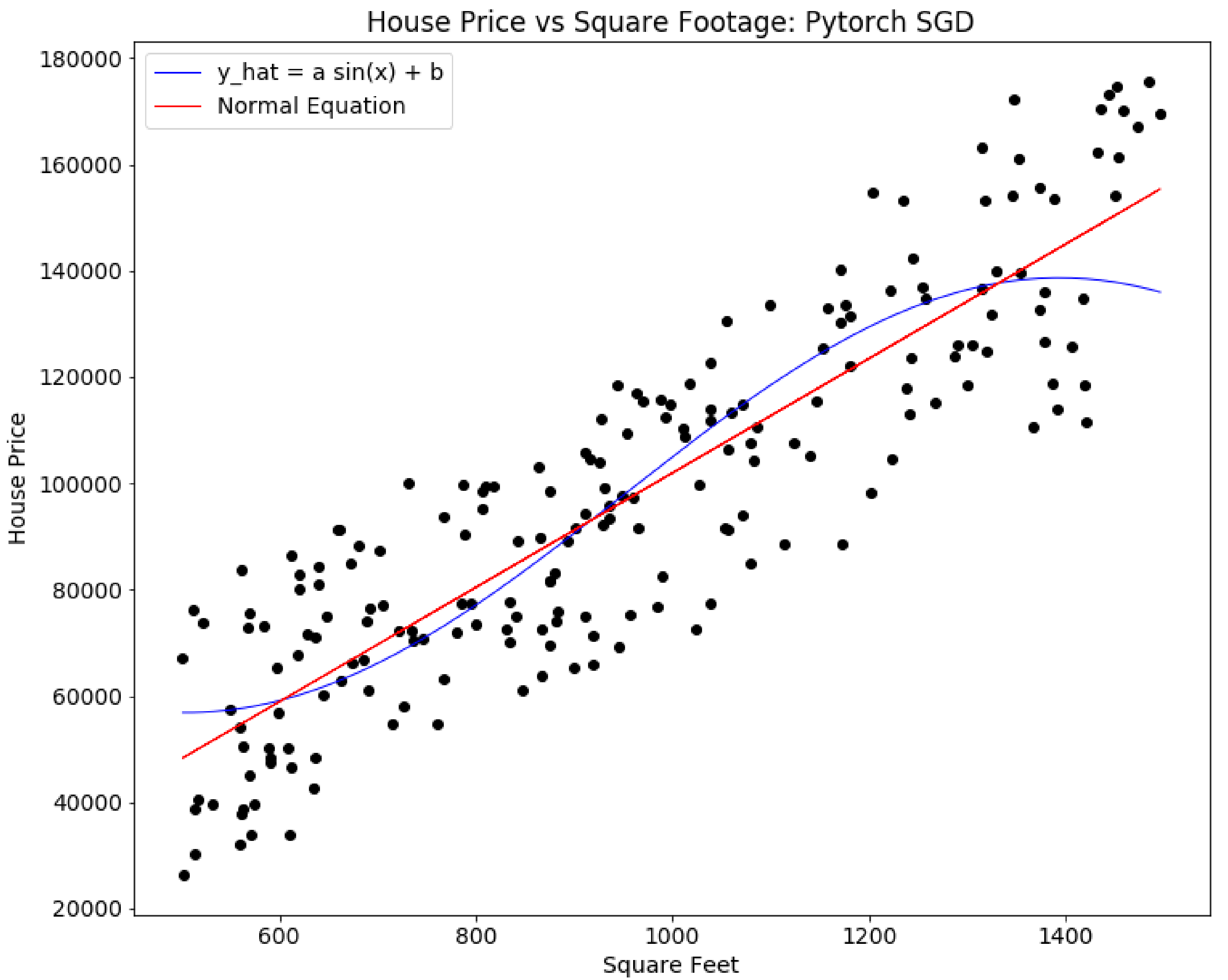
SGD: Sine Regression

```
a = Variable(torch.ones(1,1), requires_grad=True)
b = Variable(torch.ones(1,1), requires_grad=True)

optimizer = torch.optim.SGD([a, b],lr=0.0001) # # Use SGD machine-learning algorithm
loss_fn = torch.nn.MSELoss()      # Use Mean-Squared-Error loss metric

for i in range(50000):            # Take 50k steps downhill
    y_hat_2 = a*np.sin(x) + b     # Model
    loss = loss_fn(y_hat_2, y)    # Calculate MSE for this particular model
    optimizer.zero_grad()
    loss.backward()               # Calculate slope of MSE loss surface
    optimizer.step()              # Step downhill
```

SGD: Python Code



SGD: Mathematical Background

$$MSE = \frac{1}{2m}\sum_{i=1}^{m}(\hat{y}_i - y_i)^2 = \frac{1}{2m}\sum_{i=1}^{m}(ax_i + b - y_i)^2$$

Calculate the partial derivative of the loss function with respect to each of its parameters. This tells you the slope of the loss surface wrt each of parameters.

$$\frac{\partial MSE}{\partial a} = \frac{1}{m}\sum_{i=1}^{m}(ax_i + b - y_i)x_i$$

$$\frac{\partial MSE}{\partial b} = \frac{1}{m}\sum_{i=1}^{m}(ax_i + b - y_i)$$

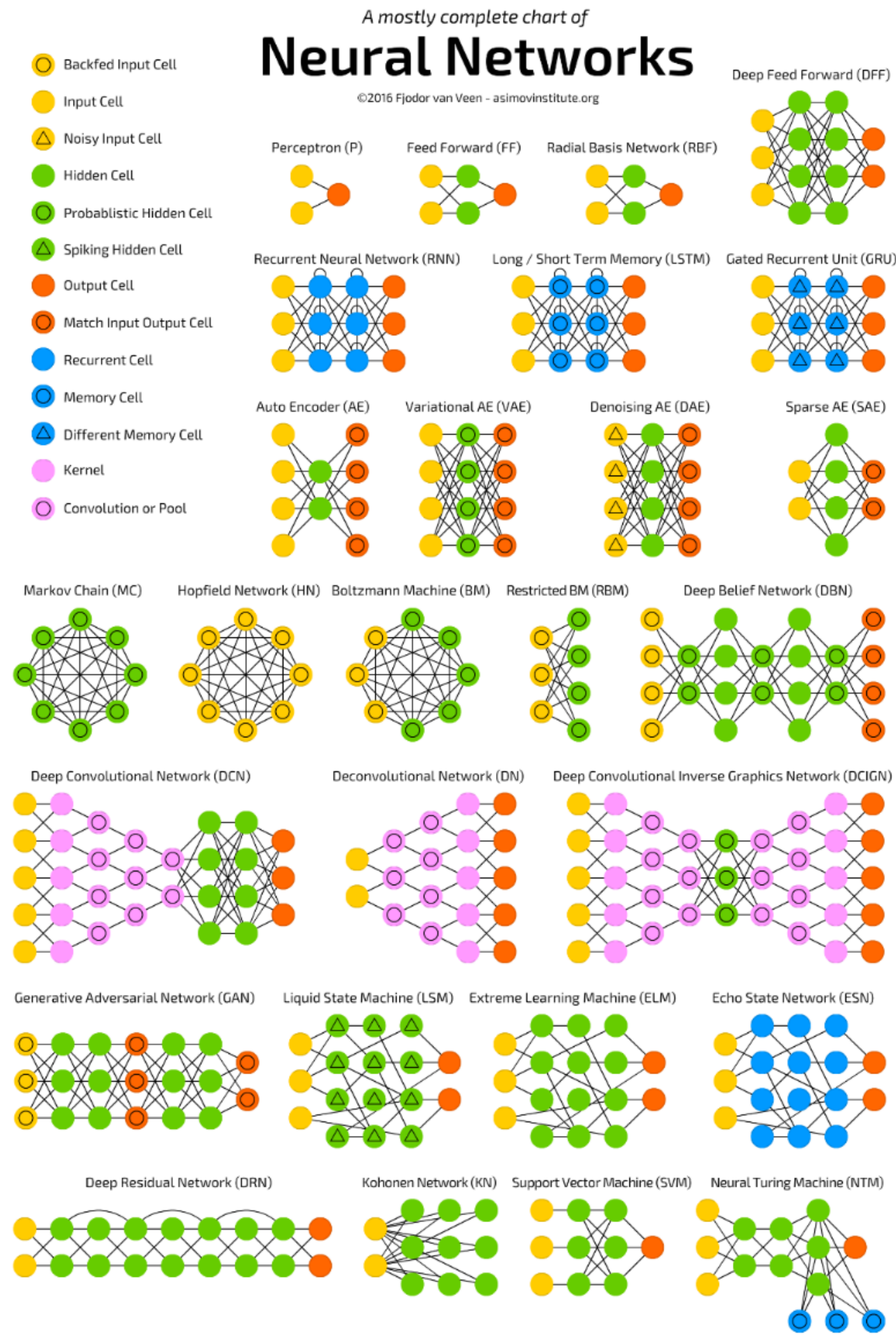
Then move towards the lowest point on the loss surface by taking small steps downslope.

- If the slope is positive, reduce the parameter.
- If the slope is negative, increase the parameter.

Benefits of SGD

- It is straightforward to calibrate predictive models
- You can build models with thousands of parameters
  - Can work on huge data sets
  - Can achieve human-level accuracy
- You can build models for all different types of data
  - Pictures
  - Videos
  - Audio
  - Text
  - Policyholder datafiles

Neural Network Models



http://www.asimovinstitute.org/neural-network-zoo/

Benefits of SGD

- It works very well in practice
  - You can choose models which are a good fit to the data
  - Rather than choosing models which you are able to fit to the data

Agenda

- What is AI?
- Regression/Classification vs Specification
- How do Neural Networks work?
- Gradient Descent Optimisation
- Why Should Actuaries be Interested?
- Examples

Jobs that pay over $100k

Job title	Average annual salary	Job title	Average annual salary
Neurologist	$217,837	Data scientist	$135,315
Psychiatrist	$194,563	Chief financial officer	$127,887
Anesthesiologist	$173,694	Android developer	$120,971
Radiologist	$168,706	Senior software engineer	$119,791
Physician	$165,391	Full stack developer	$111,709
Dentist	$157,250	Actuary	$111,474
Director of product management	$147,363	Tax manager	$108,515
Surgeon	$140,892	Director of business development	$107,789
Machine learning engineer	$137,332	Architect	$104,080
Vice president of sales	$136,071	Nurse practitioner	$103,233

Source: Indeed

indeed

Source: Indeed.com, November 2017

Why Should Actuaries Be Interested?

- Powerful new tools to solve real-world problems
  - Neural Networks for modelling big datafiles
  - Fast open-source end-to-end calculation abilities
  - Gradient Descent = general purpose solver for complex models
- The ultimate wider field?
  - Take actuarial skill-set out of actuarial department and into the real world
- Already familiar with handling data and regression modelling
- Low hanging fruit?
- Superstar salaries for top researchers
- Competition vs data scientists?

Opportunities for Insurance Companies

- Extract value from their data

- Better understanding of risks and opportunities by doing quick, novel analyses of the data

- Good models can do the same amount of work as 1000 people (at any particular task)
  - It may not be feasible for companies to hire 1000 people to perform a certain task
  - But they may be interested in getting an actuary to produce a model which can do that task
  - That model could be scaled up to be run on many computers so could do the work of say 1000 people

Opportunities for Companies

- New companies could develop massive structural advantages over incumbents?
  - E.g. Amazon have massive structural advantages over traditional retailers
  - E.g. companies who improve retention will increase market share over time

Next Steps

- Online courses on deep learning (e.g. Coursera / Udacity / FastAI)
- Learn Python (or Julia) https://www.reddit.com/r/learnpython/wiki/index
- Meetup groups
- SAI Data Analytics Subcommittee

Coursera Deep Learning Course

Jazz improvisation

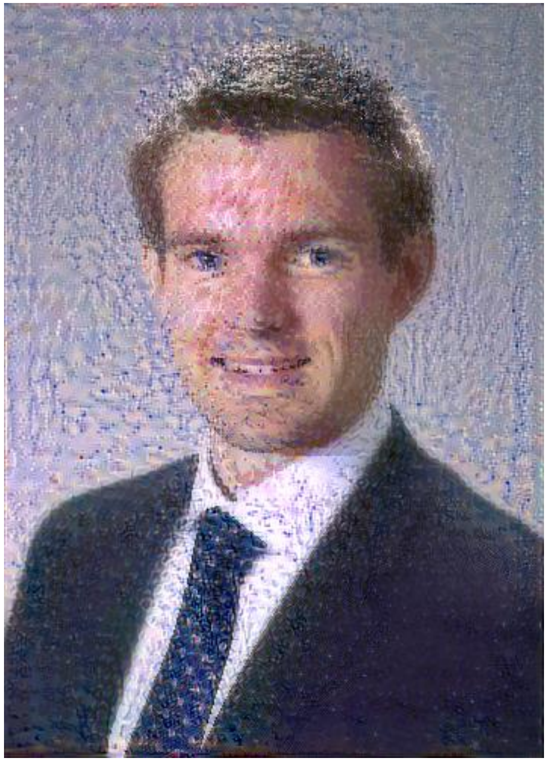
Face Recognition

Text Generation

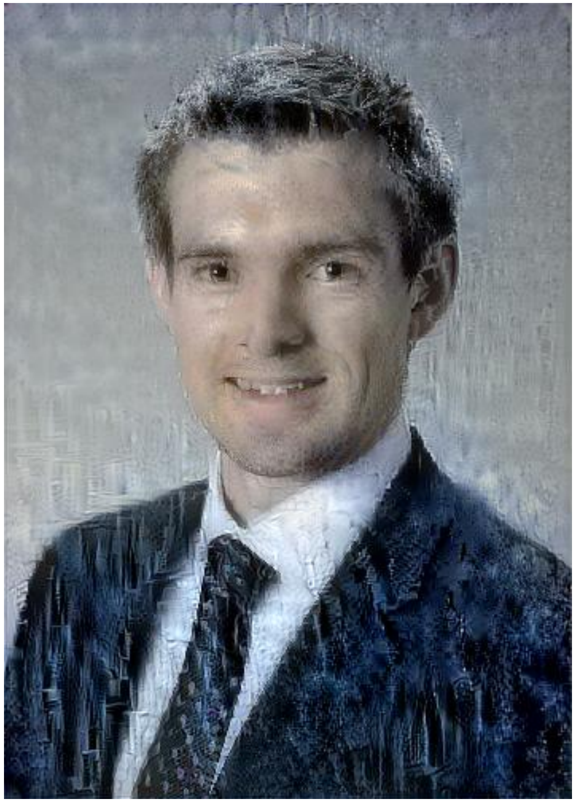
Coursera Deep Learning Course

Starry Night

Monet



Gothic



Mona Lisa

Agenda

- What is AI?
- Regression/Classification vs Specification
- How do Neural Networks work?
- Gradient Descent Optimisation
- Why Should Actuaries be Interested?
- Brainstorming

Brainstorming

What **mapping f()** do you want to discover

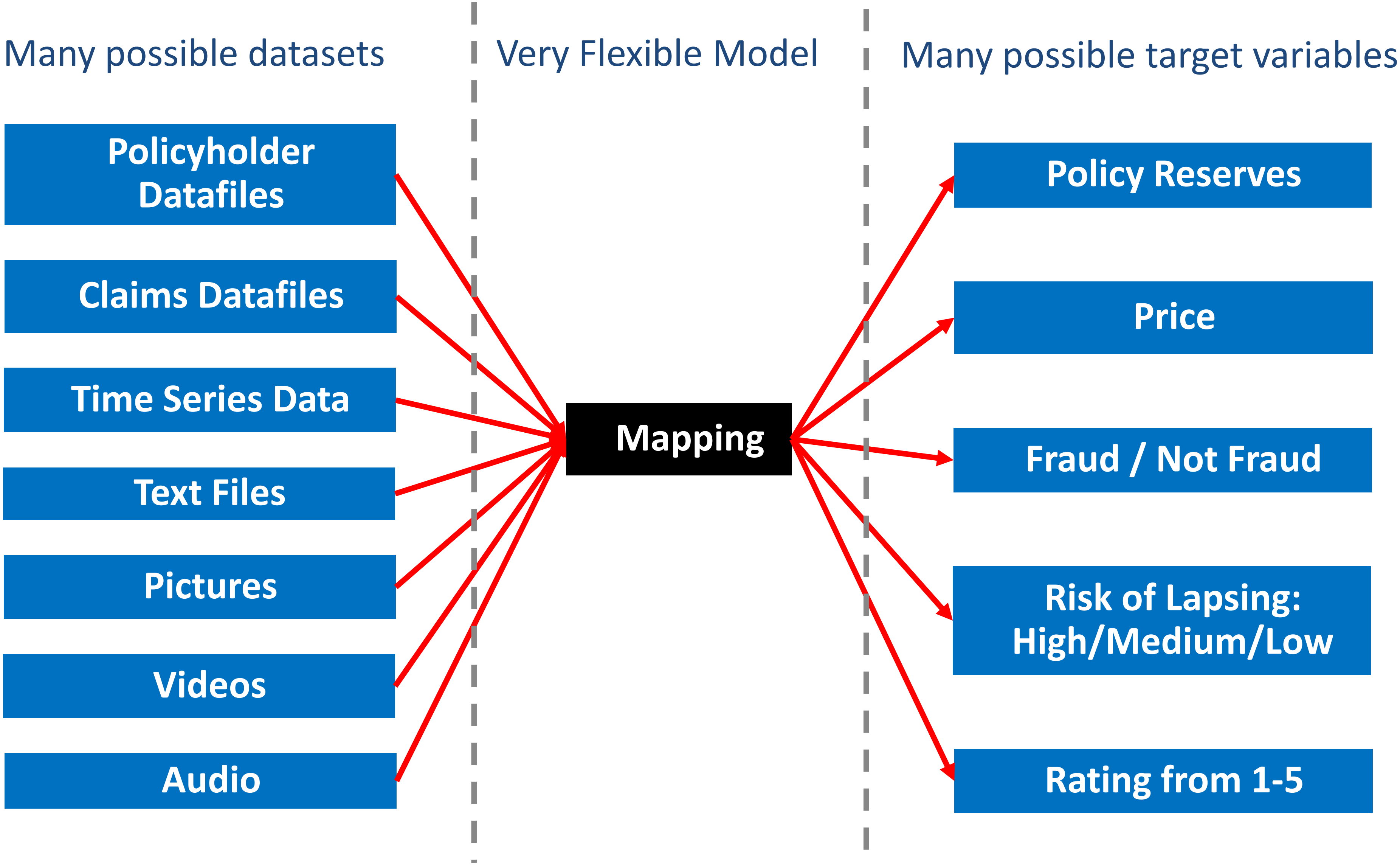
For **dataset X** and **target variable Y**

Which enables you to estimate $\widehat{Y} = f(X)$ for new or updated values of X?

Brainstorming

What output / task would you like a computer to do?

Brainstorming



General Examples

Self-Driving Cars	Speech-to-text	Fraud Detection	Customer Retention
Game Playing	Machine translation	Pricing	Proxy Models
Reducing Electricity Costs	Chatbots	Credit Risk	Call-Centre Routing
Analysing Satellite Photos	Recommender Systems	Sales Forecasting	Sentiment Analysis
Reading X-rays	Text-to-Speech	Anti-Money Laundering	Geographic Analysis

Example: Captioning

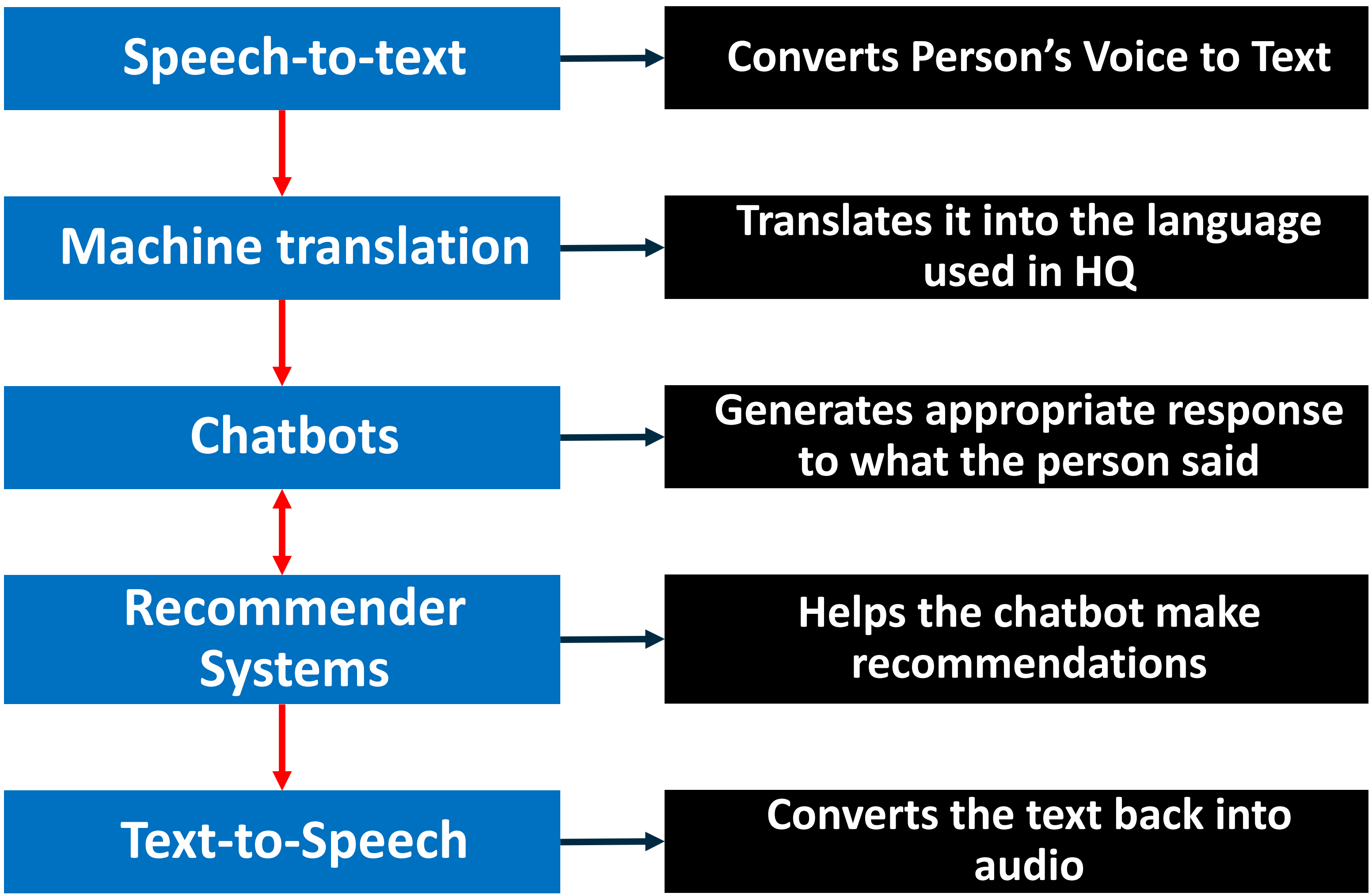
Train Model

Red dress with White Spots and Black Belt

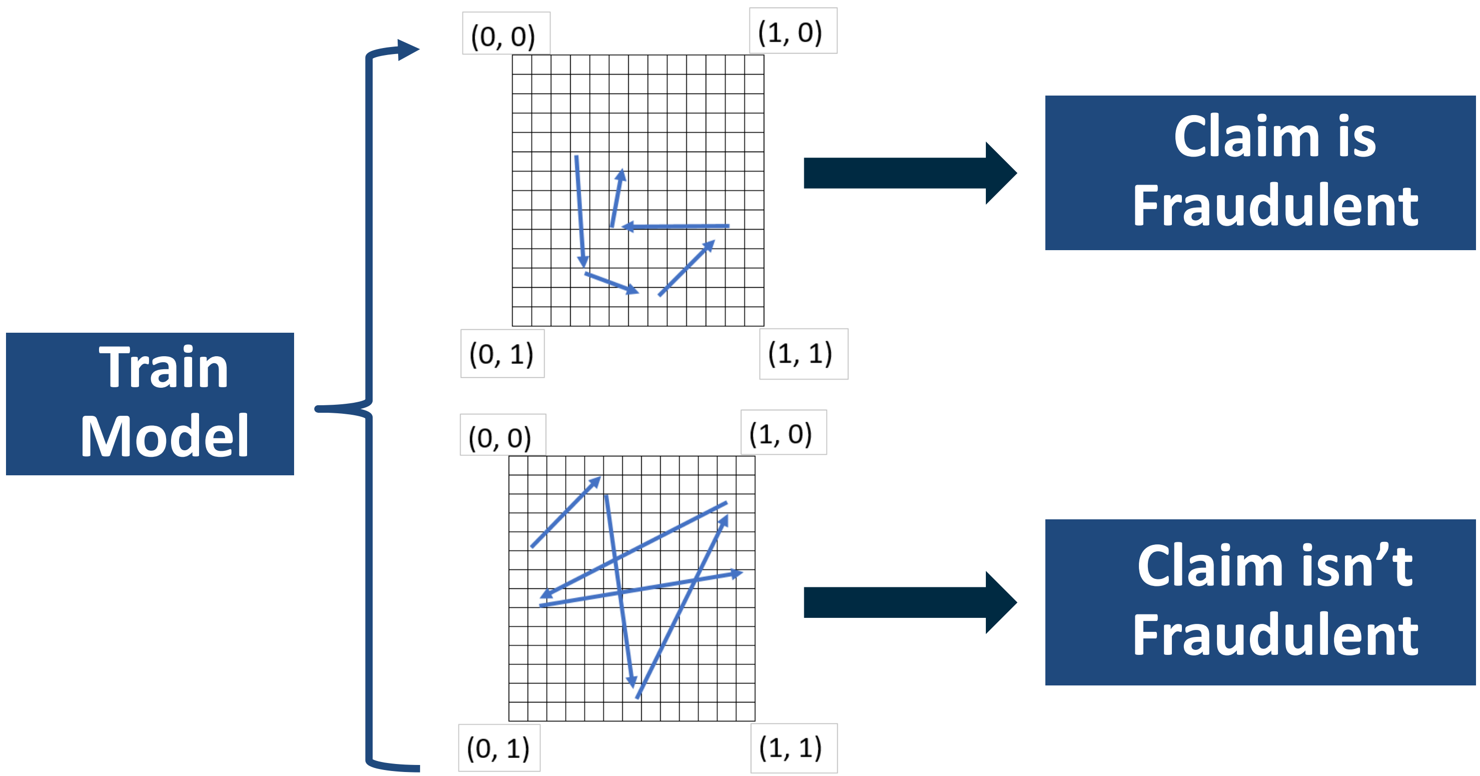
Red sweater with white stripes on arms and Gingerbread man with Christmas Hat

- In future:
  - Run thousands of pictures through the model every week
  - The model will output a caption for each picture
  - Use model output in recommender system and stock system
  - The model predicts what a human captioner would describe it as

Automated Phone Answering System

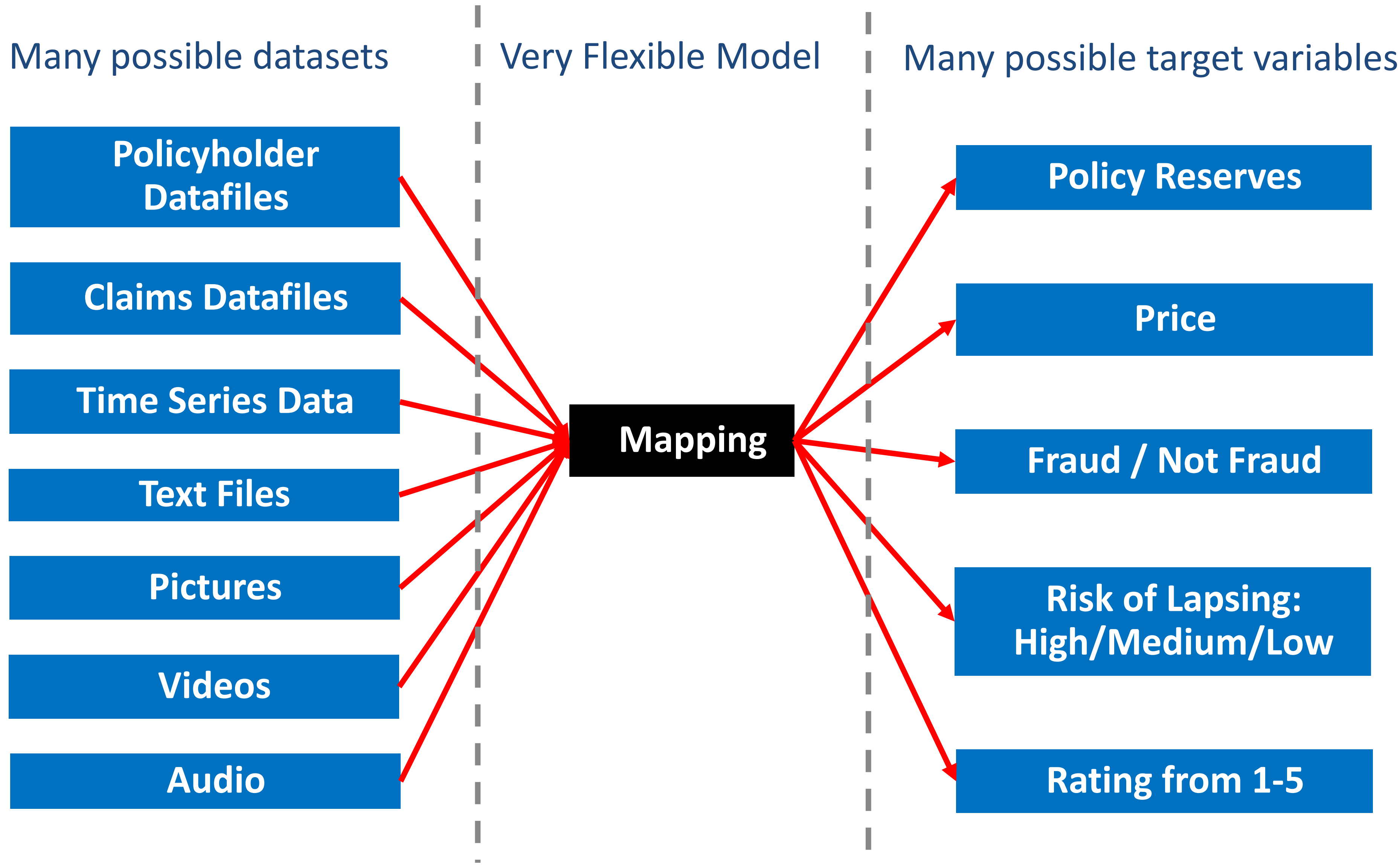


Example: Fraud Detection



- In future:
  - Record the mouse tracks for each claim
  - Run these through the model
  - The model will predict whether each incoming claim is fraudulent or non-fraudulent

Brainstorming



When to use Neural Networks

Some or all of:

- When the problem can’t be easily solved using functional specification
  - When you have noisy real-world data
- When you have lots of data
- When you have access to high-speed computing systems
- When accuracy is more important than interpretability
  - May achieve human-level accuracy but may be black-boxish
- When you need to produce results regularly and quickly

Any Questions?

- What is AI?
- Regression/Classification vs Specification
- How do Neural Networks work?
- Gradient Descent Optimisation
- Why Should Actuaries be Interested?
- Examples