

WELCOME!

FIRST SESSION

martin @ reddragon.ai

sam @ reddragon.ai

25 September 2017

WiFi : SG-Guest

Problems with Installation? **ASK!**

PLAN OF ACTION

OVERALL

- Mondays = catch-up & theory
- Wednesdays = less theory & projects
- Will monitor :
 - Attendance
 - Project work completion
 - 'Final Project'
- Wrap-up by end-Nov
- TODO : Deal with \$\$ ASAP

PLAN OF ACTION

TODAY

- This + landscape + DIY
- Intros all around
- Math
- 'Sizzle'

PLAN OF ACTION

WEDNESDAY

- Deep-and-Wide / Cross-and-Deep
- Final Project scope discussion
- Launch into a pre-defined project

PLAN OF ACTION

NEXT MONDAY

- Start with project wrap-up
- ...

ABOUT ME

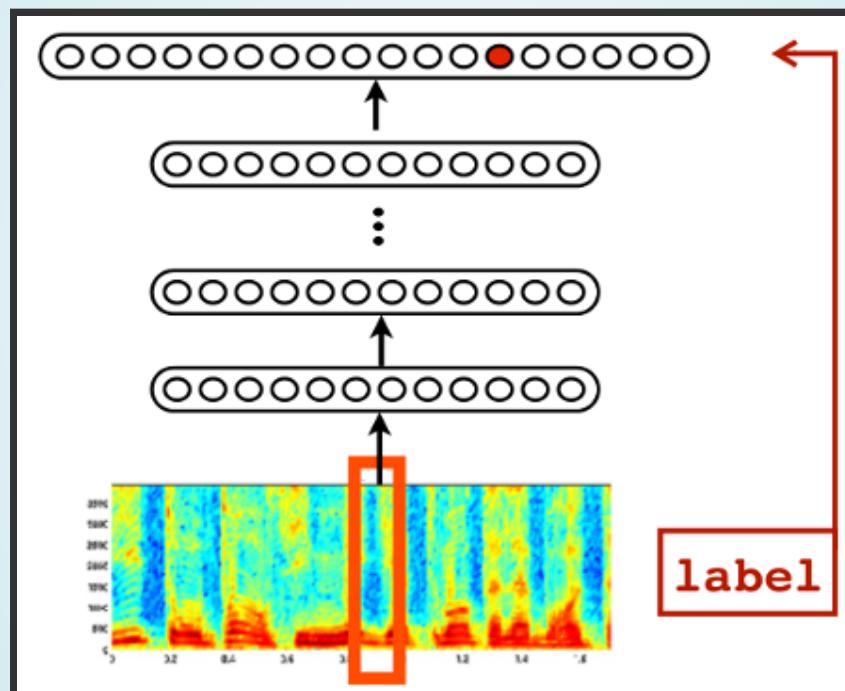
- Machine Intelligence / Startups / Finance
 - Moved from NYC to Singapore in Sep-2013
- 2014 = 'fun':
 - Machine Learning, Deep Learning, NLP
 - Robots, drones
- Since 2015 = 'serious' :: NLP + deep learning
 - & Papers...

WHAT CAN BE DONE NOW

- Speech recognition
- Language translation
- Vision :
 - Object recognition
 - Automatic captioning
- Reinforcement Learning

SPEECH RECOGNITION

Android feature since [Jellybean \(v4.3, 2012\)](#) using Cloud
Trained in ~5 days on 800 machine cluster



Embedded in phone since Android [Lollipop \(v5.0, 2014\)](#)

TRANSLATION

Google's Deep Models are on the phone



"Use your camera to translate text instantly in 26 languages"

Translations for typed text in 90 languages

HOUSE NUMBERS

Google Street-View (and ReCaptchas)



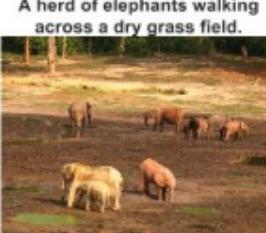
Better than human

IMAGE CLASSIFICATION



(now better than human level)

CAPTIONING IMAGES

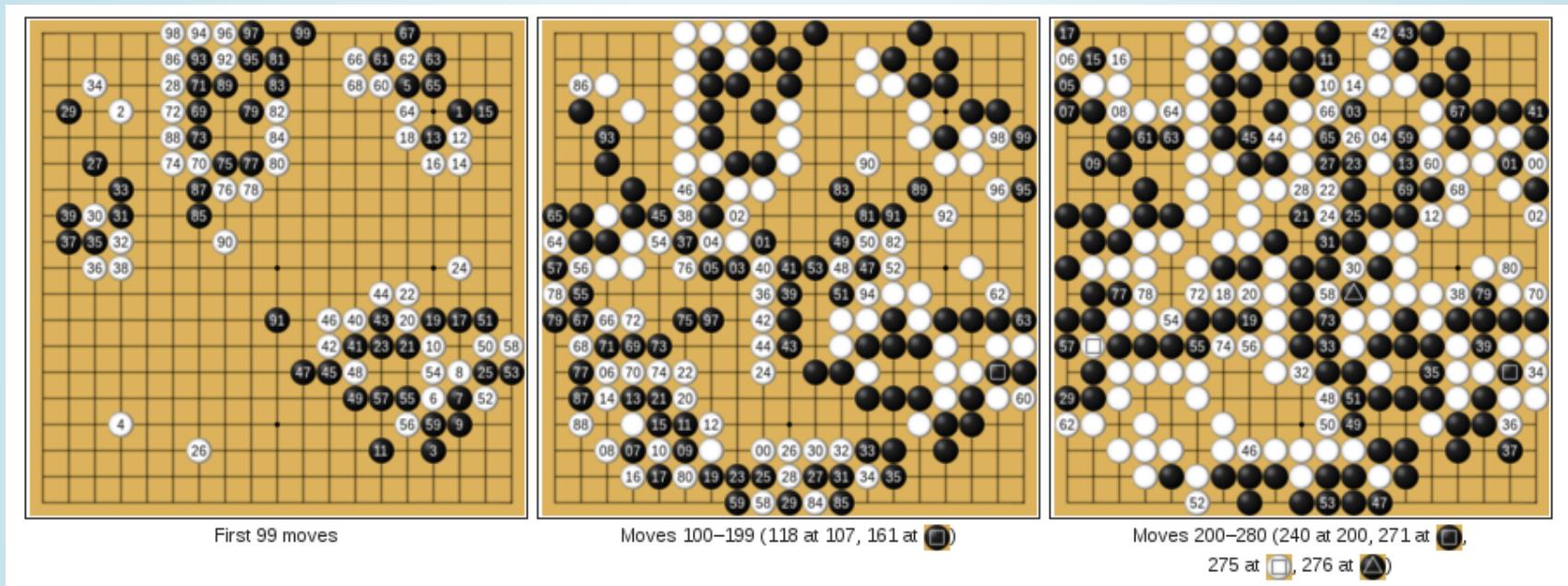
<p>A person riding a motorcycle on a dirt road.</p> 	<p>Two dogs play in the grass.</p> 	<p>A skateboarder does a trick on a ramp.</p> 	<p>A dog is jumping to catch a frisbee.</p> 
<p>A group of young people playing a game of frisbee.</p> 	<p>Two hockey players are fighting over the puck.</p> 	<p>A little girl in a pink hat is blowing bubbles.</p> 	<p>A refrigerator filled with lots of food and drinks.</p> 
<p>A herd of elephants walking across a dry grass field.</p> 	<p>A close up of a cat laying on a couch.</p> 	<p>A red motorcycle parked on the side of the road.</p> 	<p>A yellow school bus parked in a parking lot.</p> 
<p>Describes without errors</p>	<p>Describes with minor errors</p>	<p>Somewhat related to the image</p>	<p>Unrelated to the image</p>

Some good, some not-so-good

REINFORCEMENT LEARNING

Google DeepMind's AlphaGo

Learn to play Go from (mostly) self-play



"A.I. EFFECT"

- A.I. is whatever hasn't been done yet

DEEP LEARNING

- Neural Networks
- Multiple layers
- Fed with lots of Data

HISTORY

- 1980+ : Lots of enthusiasm for NNs
- 1995+ : Disillusionment = A.I. Winter (v2+)
- 2005+ : Stepwise improvement : Depth
- 2010+ : GPU revolution : Data

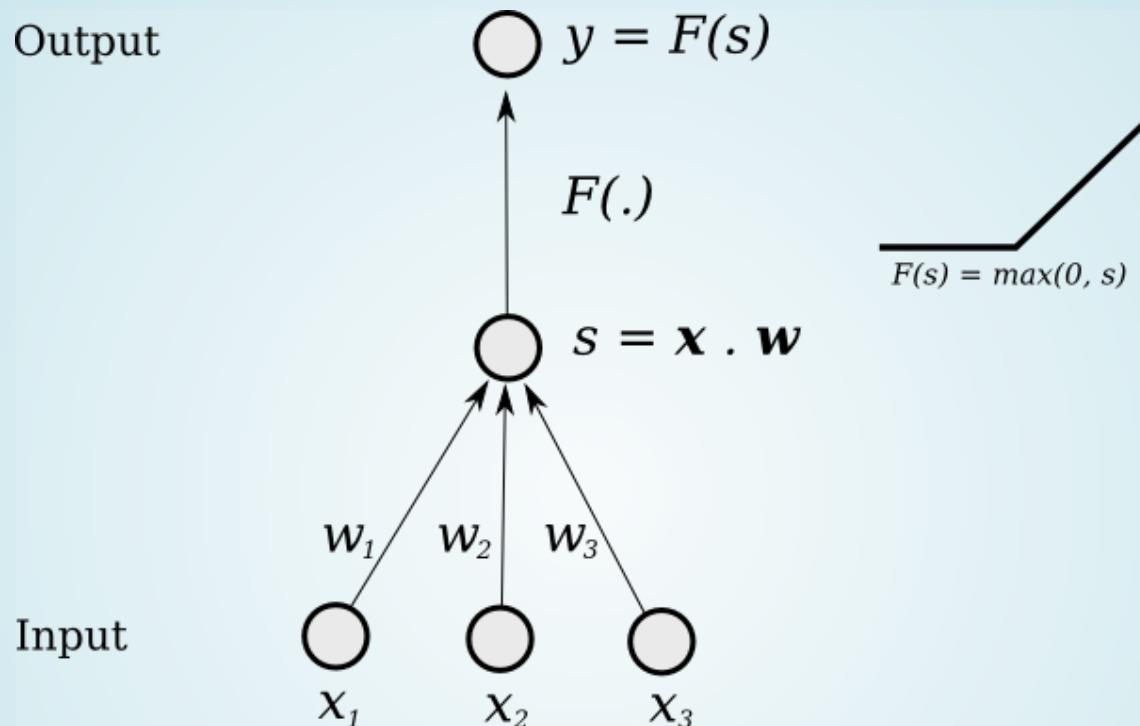
WHO IS INVOLVED

- Google - Hinton (Toronto)
- Facebook - LeCun (NYC)
- DeepMind - (Deep Bench)
- Baidu - Ng (~~Stanford~~)
- OpenAI - Karpathy (~~Stanford~~)
- ... Microsoft
- ... Apple (acquisitions), etc
- Universities, eg: Montreal (Bengio)

BASIC APPROACH

- Same as original Neural Networks since 1980s
- Simple mathematical units ...
... combine to compute a complex function

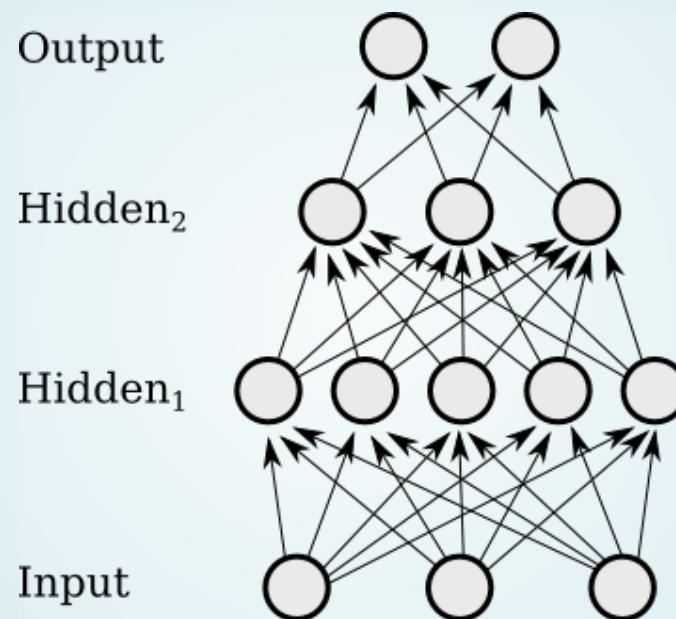
SINGLE "NEURON"



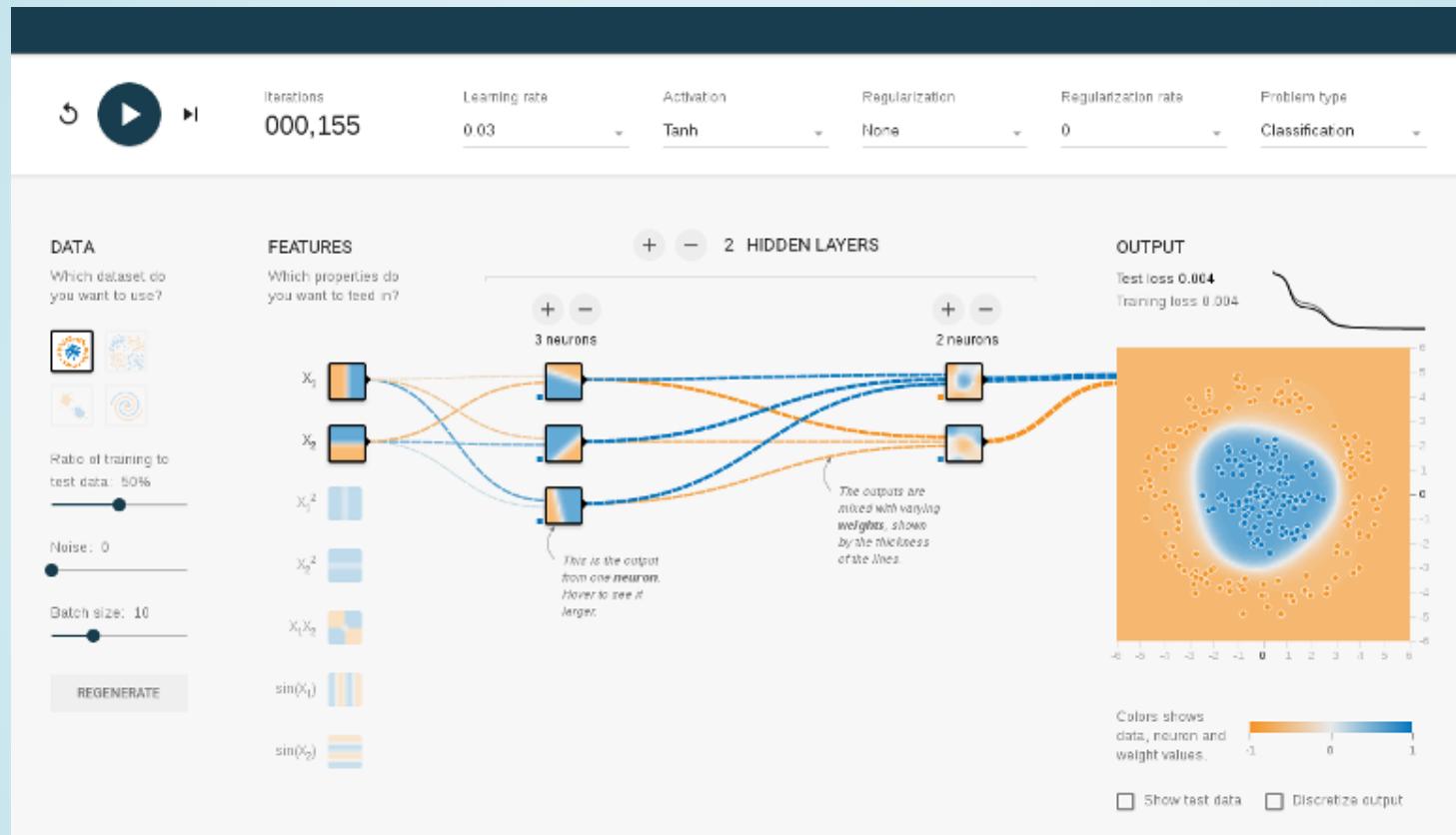
Change weights to change output function

MULTI-LAYER

Layers of neurons combine and can form more complex functions



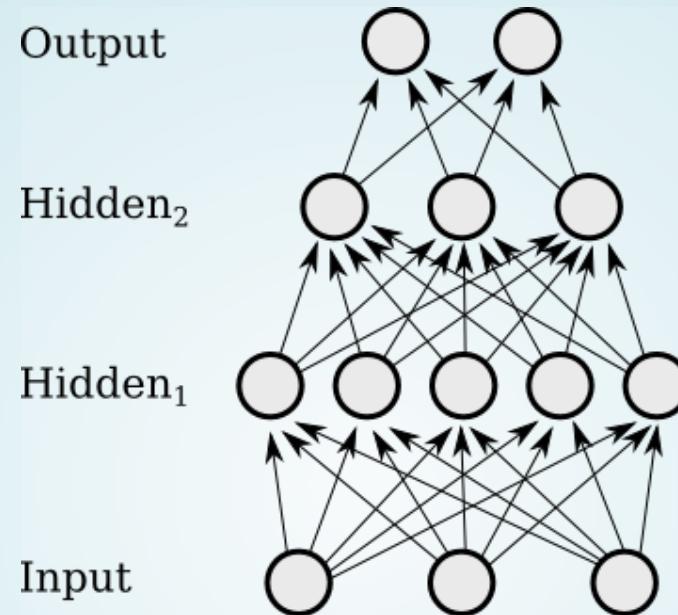
SAME THING SIDEWAYS



TERMINOLOGY

- Inputs, Outputs, Hidden Layers, Features
- Regression, Classification, Loss
- Training, Inference, Prediction

BASIC I/O



Inputs, Outputs, Hidden Layers, Features

OUR GOALS

- Regression
 - WeatherConditions → Temperature
 - HistoricalData → PxTarget
- Classification
 - WeatherConditions → Sunny/Rainy
 - HistoricalData → Buy/Sell/Hold

TWO MODES

- Development :
 - Train Model
 - Show Input and Output pairs (supervised learning)
- Production :
 - Test model : Is the output correct?
 - Infer : Fancy word for Predict
 - Predict : Use the model Output 'for real'
- Validation : check during training on extra data

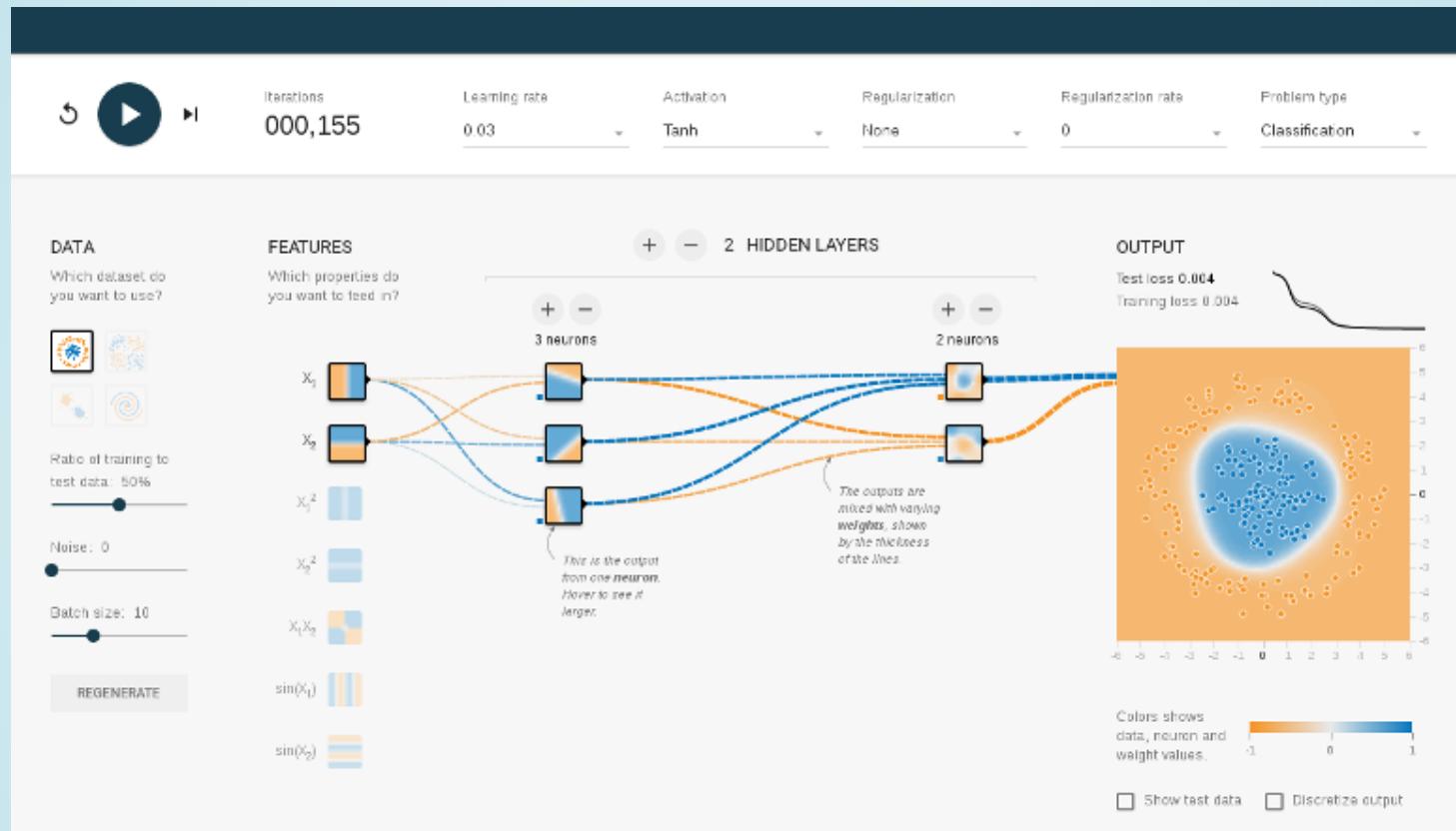
NETWORK GOALS

- Create a measure of model 'badness'
 - ... compare desired Output with model Output
- Training is a process of minimising 'badness'
 - ... which should be equivalent to our goals
- This 'badness' is called the Loss
- Minimise Loss → Improving/Training Network

MINIMISING LOSS

- Suppose one I/O pair is very bad (high Loss)
- Why was it bad?
 - Some inputs/features in the network were bad
- Try to fix the problem :
 - Pay less attention to those features; and
 - Make the feature more correct
- Fix up the weights / features little-by-little
 - To avoid breaking other examples

TENSORFLOW PLAYGROUND



WHAT'S GOING ON?

- Task : Colour the regions to capture the points
- Dataset : Choose 1 of the 4 given
- Input : Sample points
- Output : Colour of each point
- First Layer : Pre-selected features
- Training : Play button
- Loss Measure : Graph
- Epoch : 1 run through all training data

THINGS TO UNDERSTAND

- Hands-on :
 - What happens during training
 - What a single neuron can learn
 - Fixing the network iteratively
 - Features matter a lot
 - How deep networks 'create' features

THINGS TO DO

- Investigate :
 - Which datasets are hard
 - Minimal set of features
 - Minimal # of layers
 - Minimal widths
 - Effect of going less-minimal

- INTERMISSION -

MARTINANDREWS @
REDDRAGON.AI

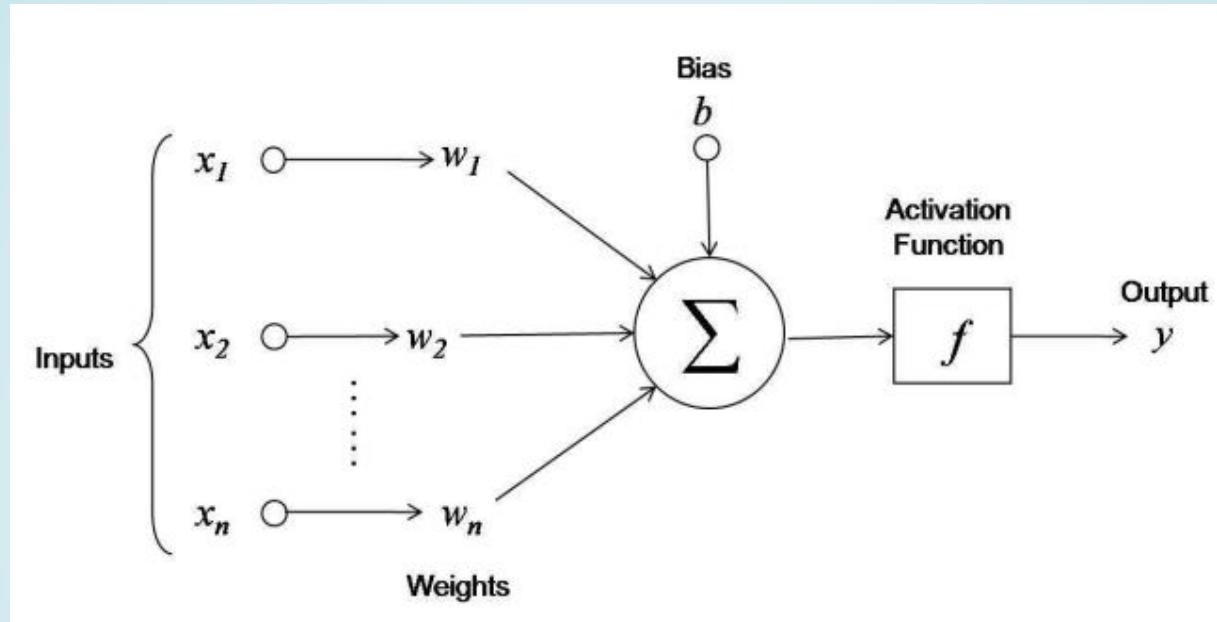
My blog : <http://mdda.net/>

GitHub : [mdda](#)

MATHEMATICS

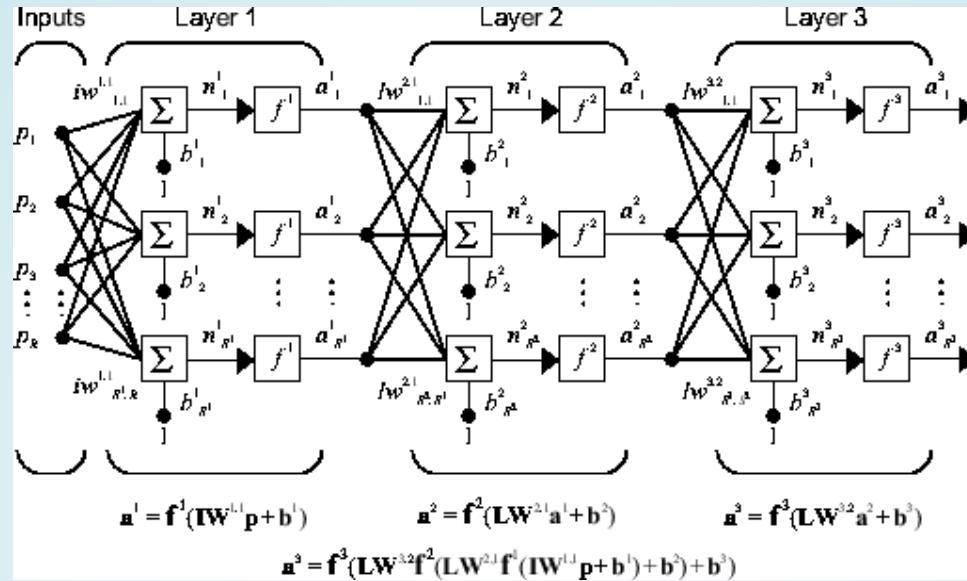
- More precise definitions
- Need to consider :
 - Weights, Bias terms
 - Non-linearities / Activations
 - Model function
 - Loss functions
 - Gradient descent

WEIGHTS, ETC



```
output = activation( ( w1*x1 + w2*x2 + ... + wn*xn ) + bias )
y = f( np.dot(ws, xs) + b )
```

LAYER-WISE



```
# Now x, y and b are vectors, w is a matrix - for a 'Dense' net
y = f( np.dot(w, x) + b )
```

ACTIVATION

Name	Plot	Equation	Derivative
Identity		$f(x) = x$	$f'(x) = 1$
Binary step		$f(x) = \begin{cases} 0 & \text{for } x < 0 \\ 1 & \text{for } x \geq 0 \end{cases}$	$f'(x) = \begin{cases} 0 & \text{for } x \neq 0 \\ ? & \text{for } x = 0 \end{cases}$
Logistic (a.k.a Soft step)		$f(x) = \frac{1}{1 + e^{-x}}$	$f'(x) = f(x)(1 - f(x))$
Tanh		$f(x) = \tanh(x) = \frac{2}{1 + e^{-2x}} - 1$	$f'(x) = 1 - f(x)^2$
ArcTan		$f(x) = \tan^{-1}(x)$	$f'(x) = \frac{1}{x^2 + 1}$
Rectified Linear Unit (ReLU)		$f(x) = \begin{cases} 0 & \text{for } x < 0 \\ x & \text{for } x \geq 0 \end{cases}$	$f'(x) = \begin{cases} 0 & \text{for } x < 0 \\ 1 & \text{for } x \geq 0 \end{cases}$
Parametric Rectified Linear Unit (PReLU) [2]		$f(x) = \begin{cases} \alpha x & \text{for } x < 0 \\ x & \text{for } x \geq 0 \end{cases}$	$f'(x) = \begin{cases} \alpha & \text{for } x < 0 \\ 1 & \text{for } x \geq 0 \end{cases}$
Exponential Linear Unit (ELU) [3]		$f(x) = \begin{cases} \alpha(e^x - 1) & \text{for } x < 0 \\ x & \text{for } x \geq 0 \end{cases}$	$f'(x) = \begin{cases} f(x) + \alpha & \text{for } x < 0 \\ 1 & \text{for } x \geq 0 \end{cases}$
SoftPlus		$f(x) = \log_e(1 + e^x)$	$f'(x) = \frac{1}{1 + e^{-x}}$

```
for fn in [ 'linear', 'sigmoid', 'tanh', 'relu', 'elu', ]:
    model.add( Dense(64, activation=fn) )
```

MODEL FUNCTION

- We can write out the network explicitly
- Express the model succinctly

```
outputs = F( inputs ) # For some complicated 'F()'  
model = Model( inputs=inputs, outputs=outputs )
```

LOSS FUNCTIONS

Regression:

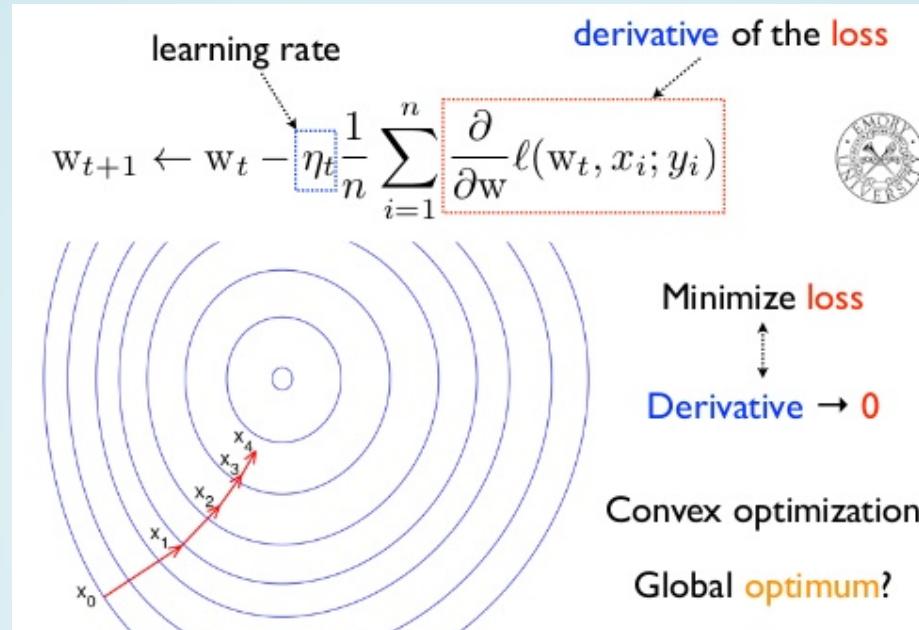
$$R(\theta) = \sum_{k=1}^K \sum_{i=1}^N (y_{ik} - f_k(x_i))^2.$$

Classification: cross-entropy (deviance)

$$R(\theta) = - \sum_{i=1}^N \sum_{k=1}^K y_{ik} \log f_k(x_i).$$

```
model_regression.compile( loss='mean_squared_error' )
model_classifier.compile( loss='categorical_crossentropy' )
```

GRADIENT DESCENT

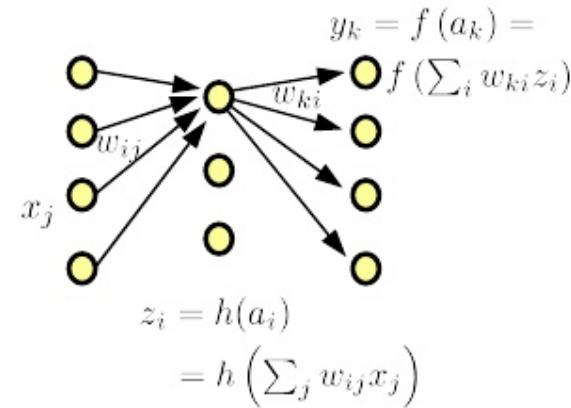


Need derivatives of Loss w.r.t *every single weight*

BACKPROPAGATION

For hidden layers, we apply the chain rule

$$\begin{aligned}\delta_i &= \frac{\partial E_n}{\partial a_i} = \sum_k \frac{\partial E_n}{\partial a_k} \frac{\partial a_k}{\partial a_i} \\ &= \sum_k \delta_k \frac{\partial a_k}{\partial a_i} = \sum_k \delta_k \frac{\partial a_k}{\partial z_i} \frac{\partial z_i}{\partial a_i} \\ &= \sum_k \delta_k w_{ki} \frac{\partial z_i}{\partial a_i} = h'(a_i) \sum_k w_{ki} \delta_k\end{aligned}$$

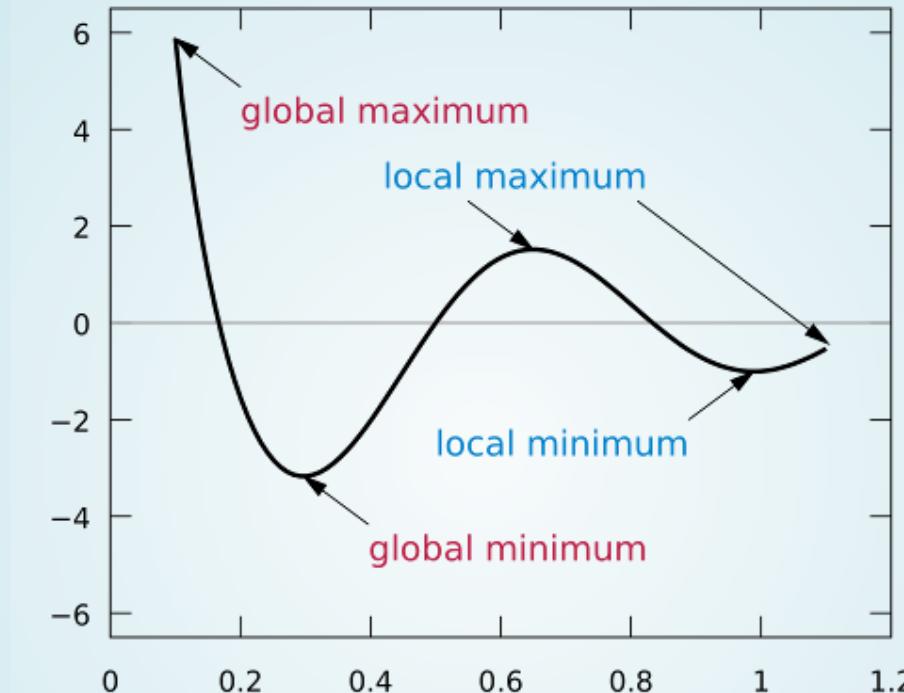


But after a forward pass, we can push the error backwards...

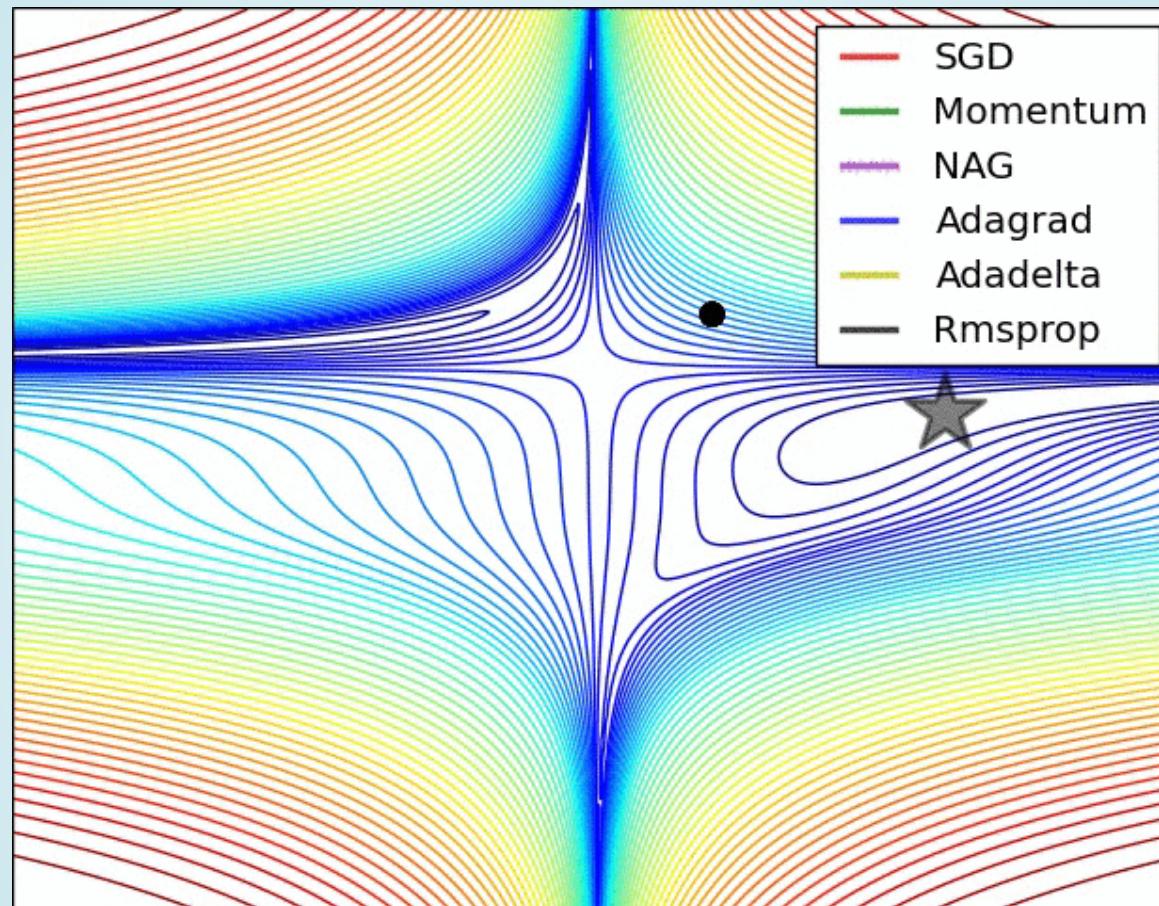
PROBLEM?

- No! : Frameworks do this for us
- Calculus is not necessary
- Better to understand failure modes...

MINIMA TYPES

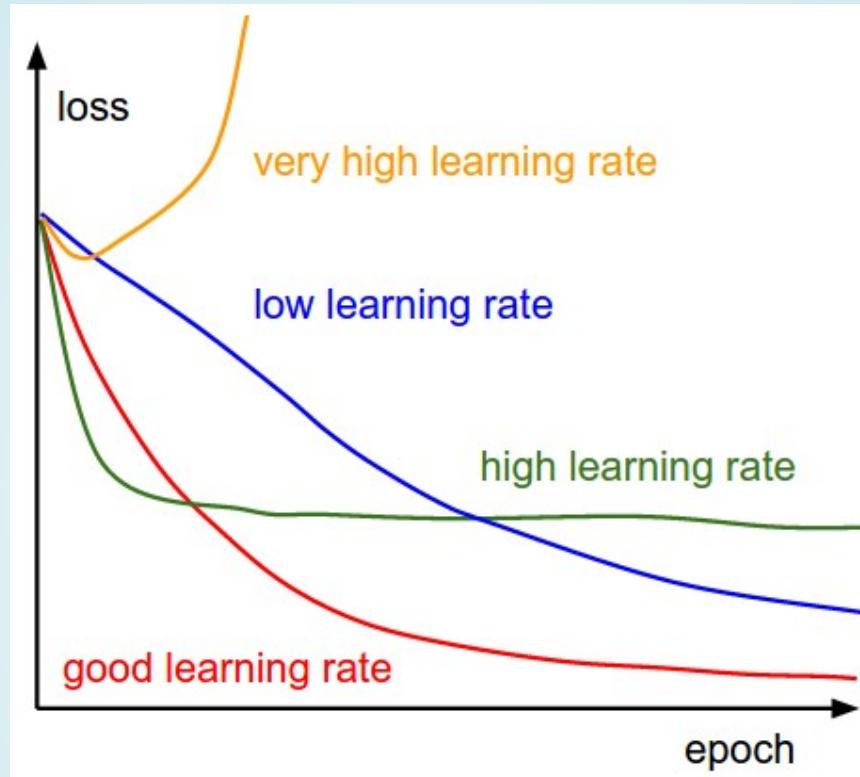


OPTIMISERS



Credit to Alec Radford

LEARNING RATES



TENSORFLOW DEEP LEARNING MEETUP GROUP

- Next Meeting = early-Nov-2017
 - Hosted by Google
- Typical Contents :
 - Talk for people starting out
 - Something from the bleeding-edge
 - Lightning Talks
- MeetUp.com / TensorFlow-and-Deep-Learning-Singapore

PYTORCH DEEP LEARNING MEETUP GROUP

- Next Meeting = 19-Oct-2017
 - Hosted by Facebook
- Typical Contents :
 - ~~Talk for people starting out~~
 - Things from the bleeding-edge
 - Lightning Talks
- MeetUp.com / PyTorch-and-Deep-Learning-Singapore

- QUESTIONS -

MARTINANDREWS@
REDDRAGON.AI

My blog : <http://mdda.net/>

GitHub : [mdda](#)