Deep Learning Techniques and Applications

Georgiana Neculae

The University of Manchester

Outline

1. Why Deep Learning?
2. Applications and specialized Neural Networks
3. Neural Networks basics and training
4. Potential issues
5. Preventing overfitting
6. Research directions
7. Implementing your own!

The University of Manchester

Why Deep Learning?

Nvidia unveils deep learning supercomputer

When deep learning becomes the game changer

Let's take the example of America's traditional pastime: baseball. New York University Professor Claudio Silva and MLB Advanced Media consultant Carlos Dietrich have developed a metrics engine that tracks each movement of every player and the ball throughout a game. The system takes in all this data and then identifies patterns that can help coaches manage players, plan strategy, and equips them with the ability

Google's DeepMind wins historic Go contest 4-1

Intel debuts a deep-learning AI chip to battle Nvidia

In what looks like a repeat of its loss to Qualcomm on smartphones, Intel has lagged graphics chip (GPU) maker Nvidia in the artificial intelligence revolution.

Clever Camera App Uses Deep Learning to Perfectly Retouch Your Photos Before You Take Them

[1,2,3,4,5]

The University of Manchester

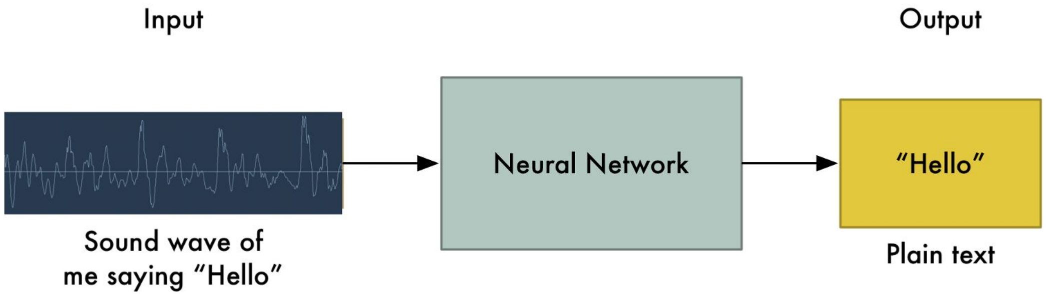
Why is it important?

Impressive performance on what was perceived as exclusively human tasks:

- Playing games
- Artistic creativity
- Verbal communication
- Problem solving

Applications

Speech Recognition

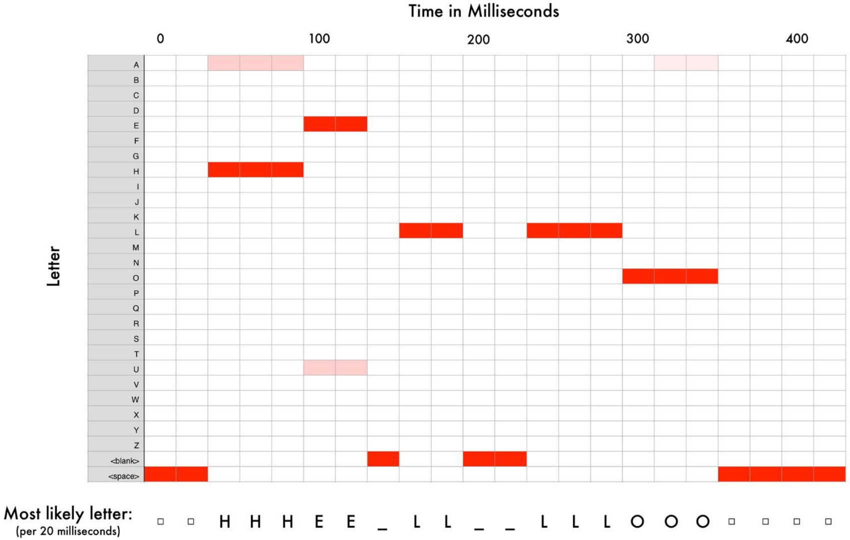


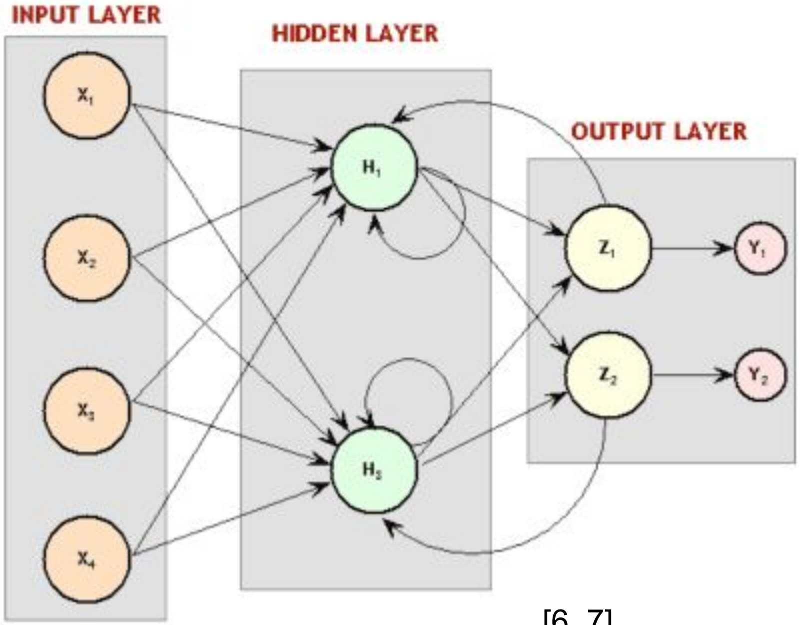
[6]

- Aim: Input speech recordings and receive text.
- Why? (translation, AI assistants, automatic subtitles)
- Challenges come from the differences between pronunciations:
  - Intonation
  - Accent
  - Speed
  - Cadence or inflection

Recurrent Neural Networks (RNNs)

- Make use of internal memory to predict the most likely future sequence based on what they have seen so far





[6, 7]

MANCHESTER 1824
The University of Manchester

WaveNet

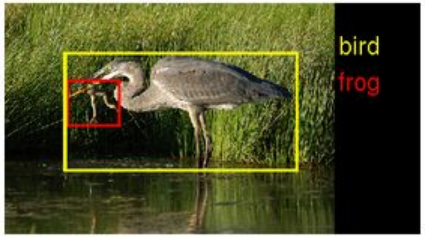
- Generates speech that sounds more natural than any existing techniques
- Also used to synthesize and generate music

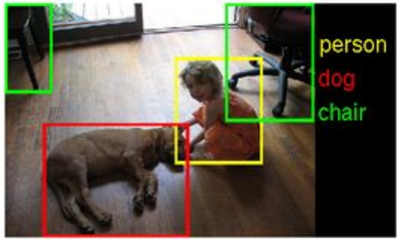
https://deepmind.com/blog/wavenet-generative-model-raw-audio/

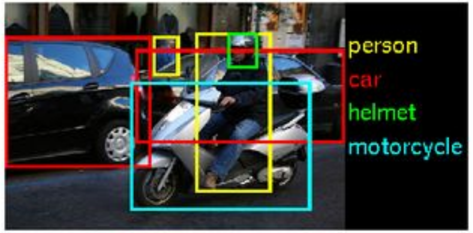
Object Detection and Recognition

- Why? (face detection for cameras, counting, visual search engine)

- What features are important when learning to understand an image?







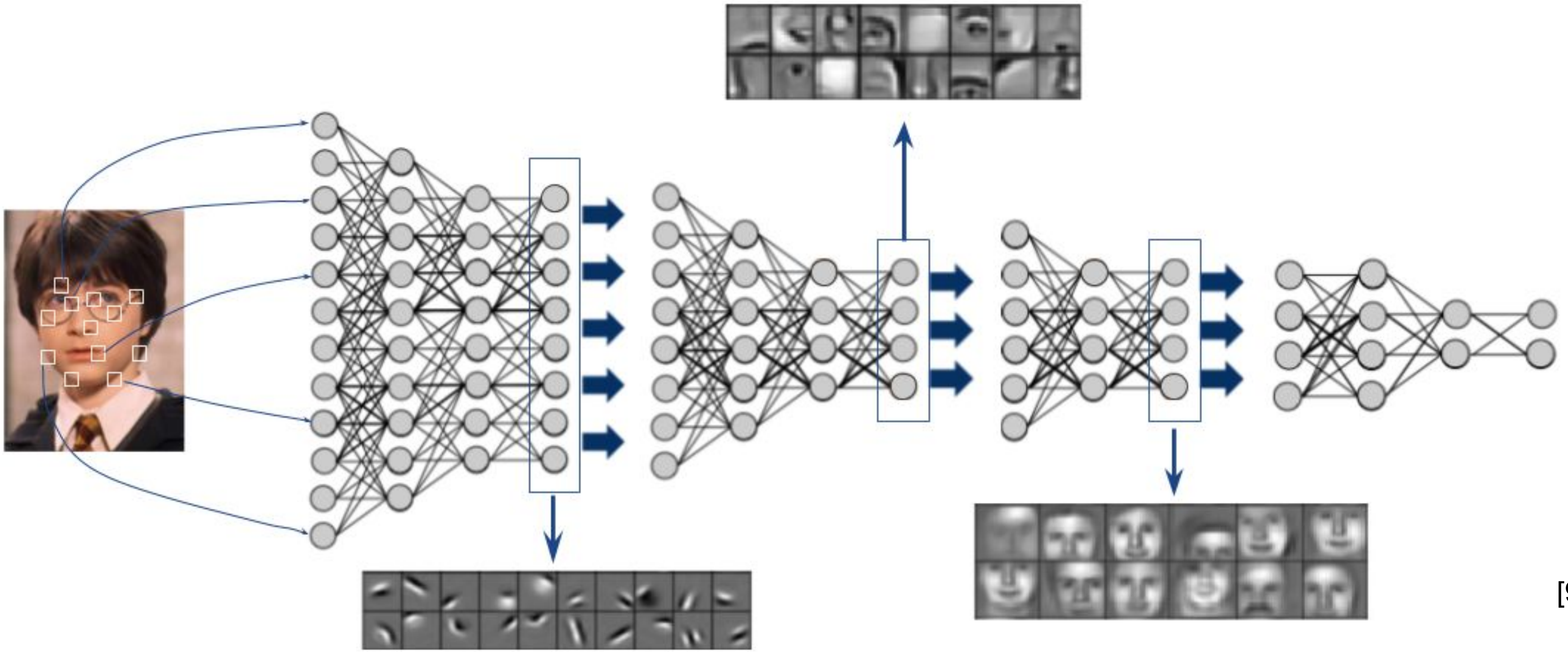
[8]

Object Detection and Recognition

- Difficulty arises from:
  - Multiple objects can be identified in a photo
  - Objects can be occluded by environment
  - Object of interest could be too small
  - Same class examples could be very different

Convolutional Neural Networks (CNNs)

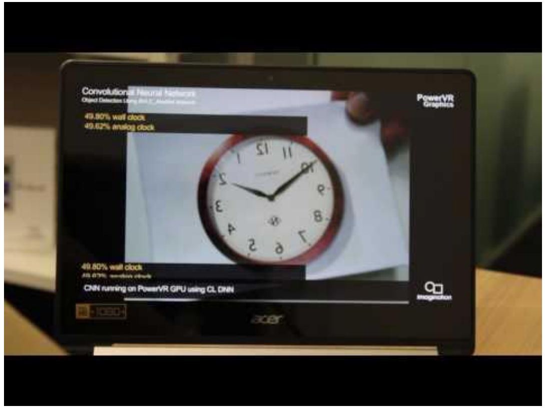
The University of Manchester



[9]

The University of Manchester

Object Recognition



Object Recognition



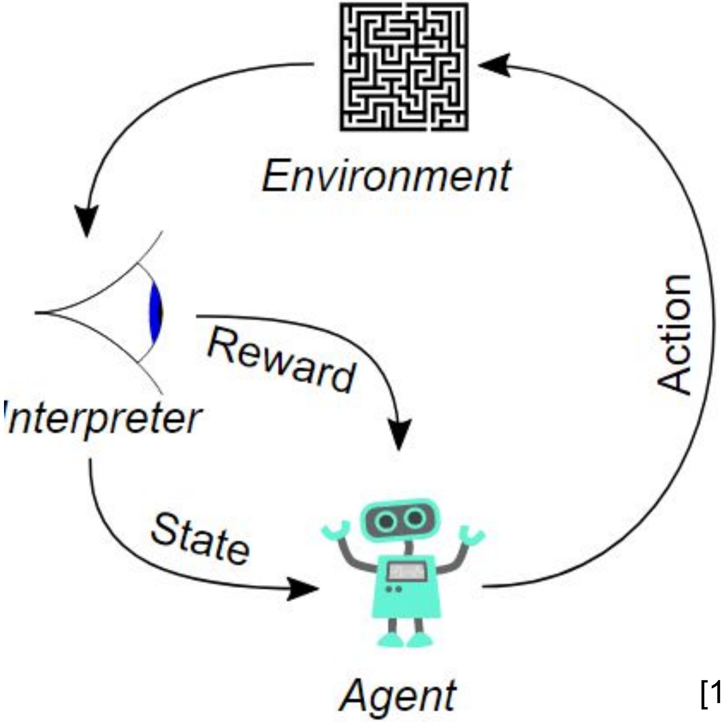



http://extrapolated-art.com/

https://deepdreamgenerator.com/feed

Reinforcement Learning

- Learning is done through trial-and-error, based on rewards or punishments
- Agents independently develop successful strategies that lead to the greatest long-term rewards
- No hand engineered features or domain heuristics are provided, the agents being capable to learn directly from raw inputs

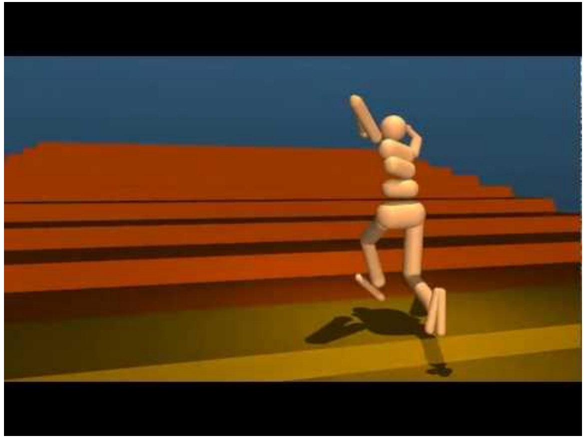


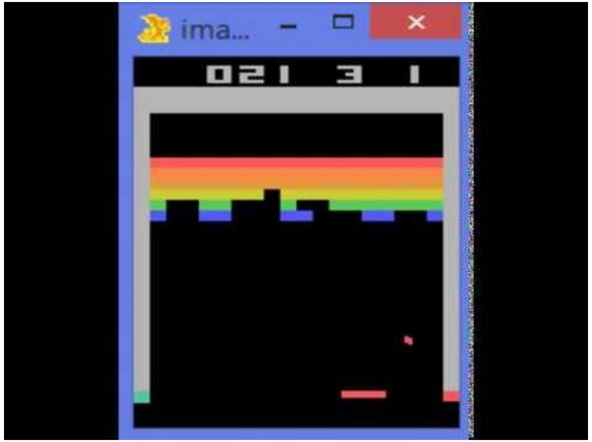
[10]

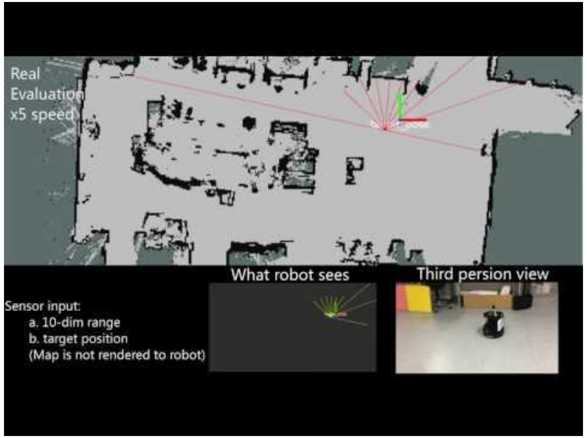
Reinforcement Learning

AlphaGo, a deep neural network trained using reinforcement learning, defeated Lee Sedol (the strongest Go player of the last decade) by 4 games to 1.

https://deepmind.com/blog/deep-reinforcement-learning/







Neural Networks Basics

The University of Manchester

Perceptron

"the embryo of an electronic computer that [the Navy] expects will be able to **walk, talk, see, write**, reproduce itself and be conscious of its existence"

Frank Rosenblatt, 1957

Perceptron to Logistic Regression (recap)

$$f(\mathbf{x}) = \mathbf{w}^T\mathbf{x} - t,$$

Perceptron

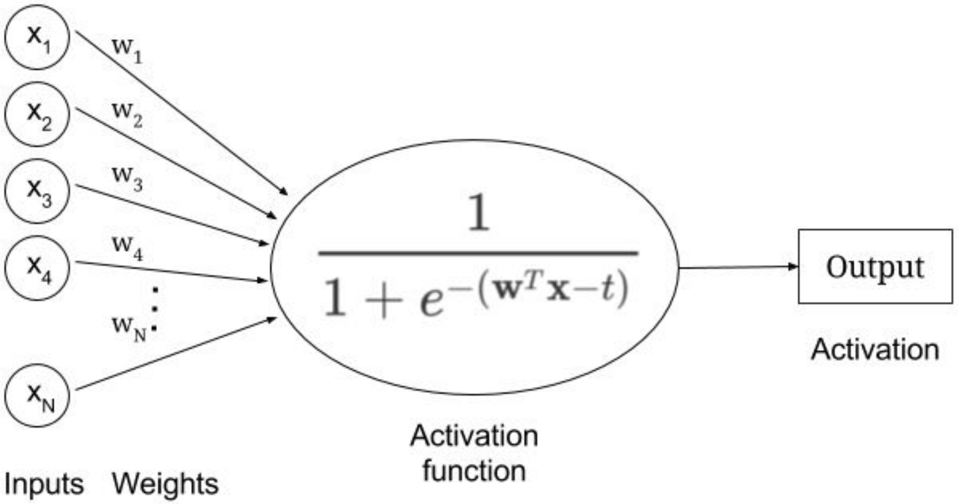
$$f(\mathbf{x}) = \frac{1}{1 + e^{-(\mathbf{w}^T\mathbf{x} - t)}}$$

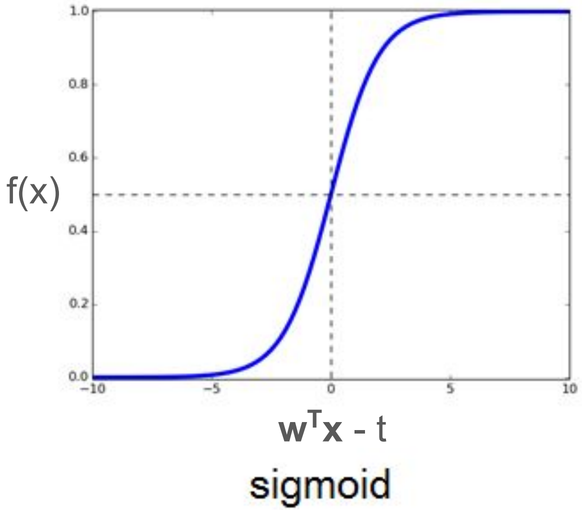
Logistic Regression

[11]

Logistic Regression (recap)

- Linear model capable of solving 2 class problems
- Uses the Sigmoid function to scale the output between [0,1]

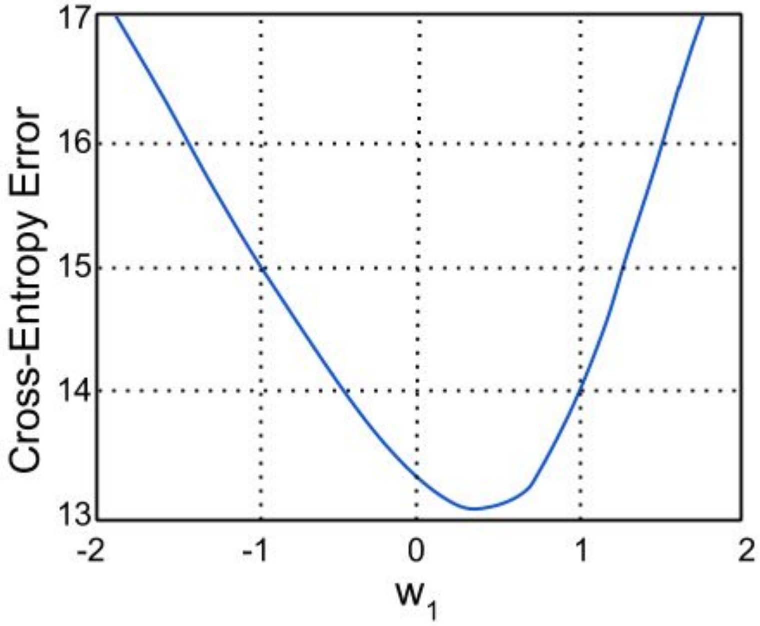




Logistic Regression (recap)

Uses the Log-loss function (cross entropy) to minimize the error:

$$E = -\sum_{i=1}^{N}\{y_i \log f(x_i) + (1-y_i)\log(1-f(x_i))\}$$



The University of Manchester

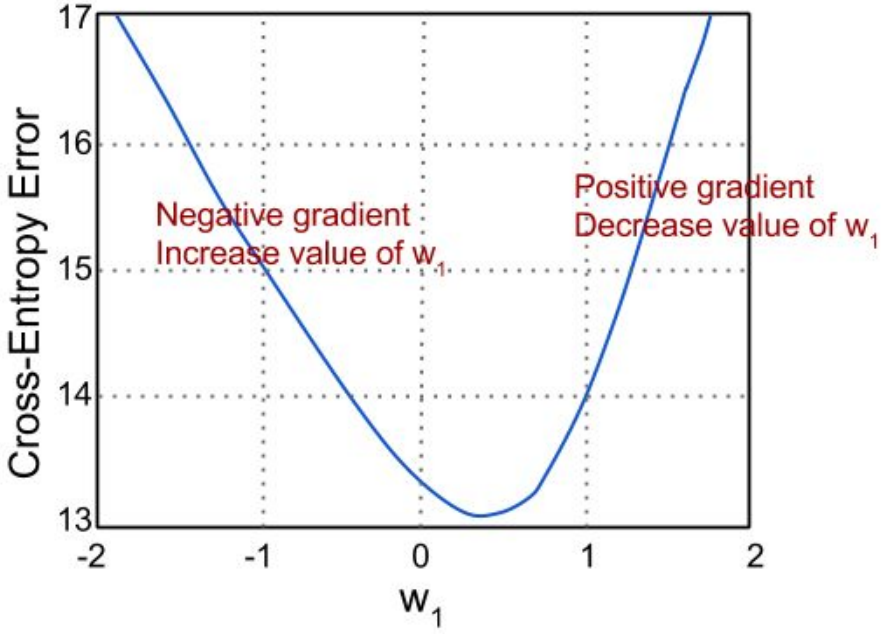
Gradient Descent (recap)

Update rule:

Update parameters in the negative direction of the gradient.

Negative gradient ➡ Increase value of w_1

Positive gradient ➡ Decrease value of w_1

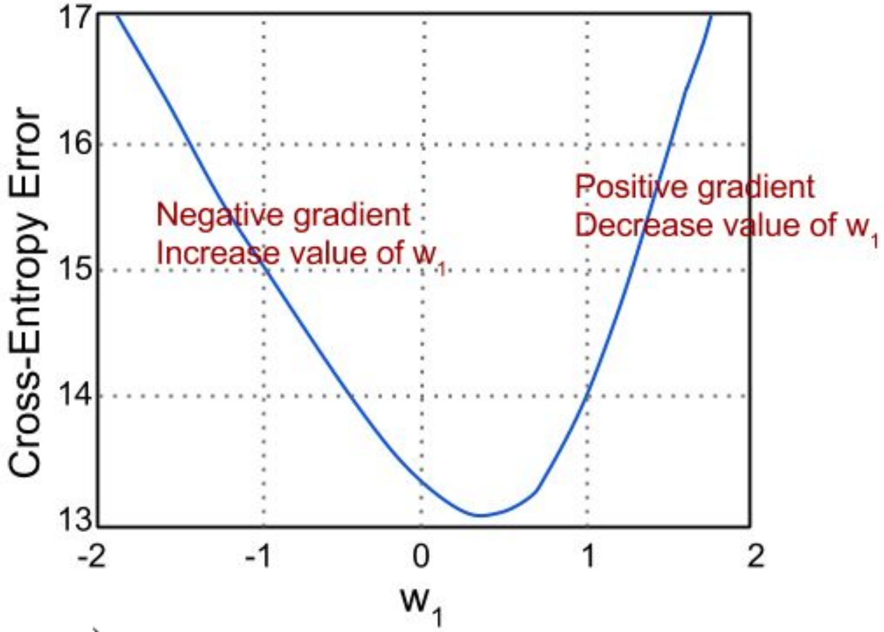


Gradient Descent (recap)

Log-loss function:

$$E = -\sum_{i=1}^{N}\{y_i \log f(x_i) + (1-y_i)\log(1-f(x_i))\}$$

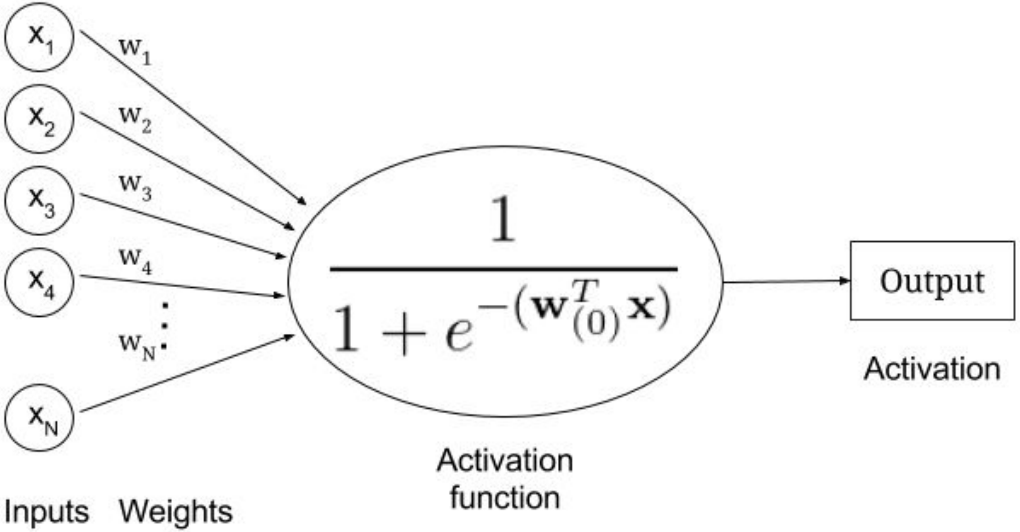
Gradient is given by the partial derivative with respect to parameter w_i:



$$\frac{\partial E}{\partial w_j} = \frac{\partial E}{\partial f(x)} \frac{\partial f(x)}{\partial(\mathbf{w}^T\mathbf{x}-t)} \frac{\partial(\mathbf{w}^T\mathbf{x}-t)}{\partial w_j} = -\sum_i (f(x_i)-y_i)x_{ij}$$

Gradient Descent

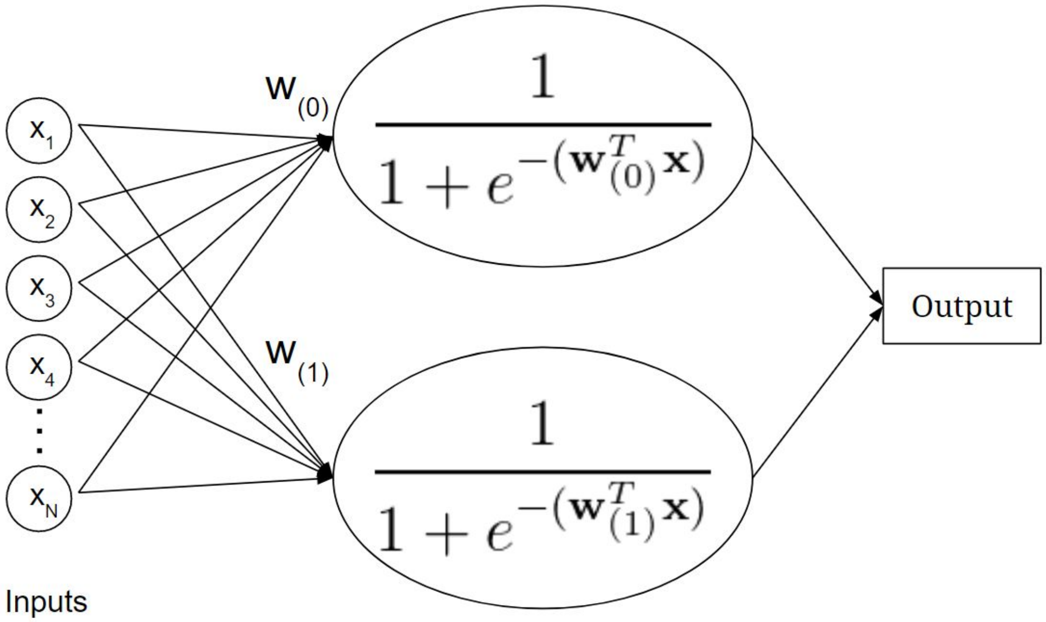
Gradient is given by the partial derivative with respect to parameter w_i:



$$\frac{\partial E}{\partial w_j} = \frac{\partial E}{\partial f(x)} \frac{\partial f(x)}{\partial(\mathbf{w}^T\mathbf{x} - t)} \frac{\partial(\mathbf{w}^T\mathbf{x} - t)}{\partial w_j} = -\sum_i (f(x_i) - y_i) x_{ij}$$

Gradient Descent

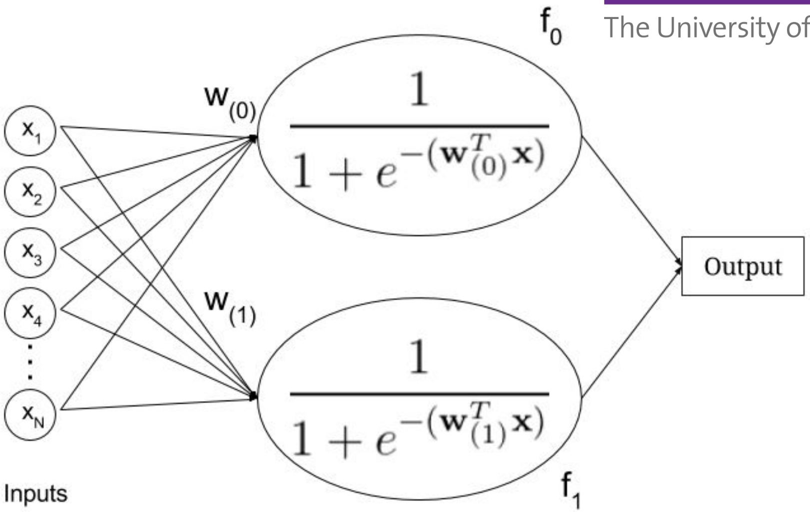
- What if we add another unit (neuron)
- How do we update the parameters?



Gradient Descent

Gradient is computed in the same way.

How do we combine the outputs of the two neurons?

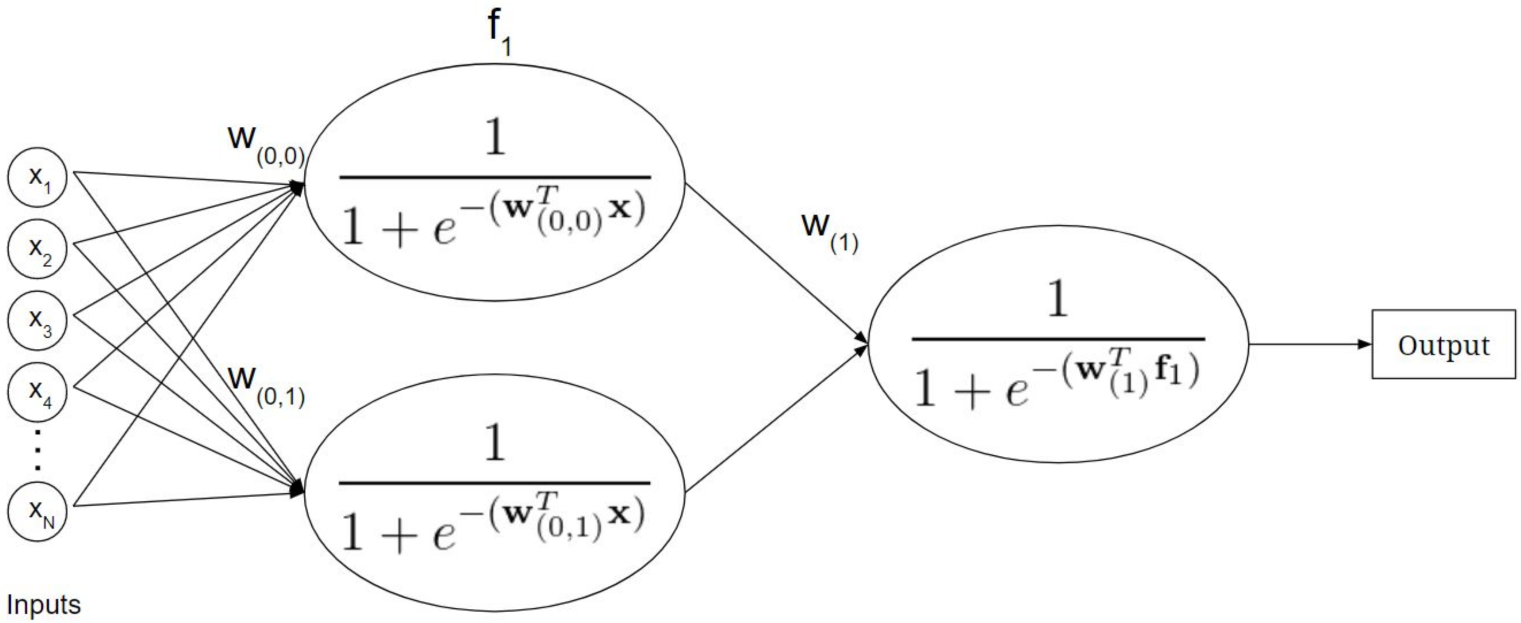


$$\frac{\partial E}{\partial w_{(0)j}} = \frac{\partial E}{\partial f_0(x)} \frac{\partial f_0(x)}{\partial(\mathbf{w}_{(0)}^T\mathbf{x} - t)} \frac{\partial(\mathbf{w}_{(0)}^T\mathbf{x} - t)}{\partial w_{(0)j}} = -\sum_i (\boxed{f_0(x_i)} - y_i)x_{ij}$$

$$\frac{\partial E}{\partial w_{(1)j}} = \frac{\partial E}{\partial f_1(x)} \frac{\partial f_1(x)}{\partial(\mathbf{w}_{(1)}^T\mathbf{x} - t)} \frac{\partial(\mathbf{w}_{(1)}^T\mathbf{x} - t)}{\partial w_{(1)j}} = -\sum_i (\boxed{f_1(x_i)} - y_i)x_{ij}$$

Multi-layer Perceptron

- Two neurons can only be combined by using another neuron:



Error Function

- Regression (network predicts real values):

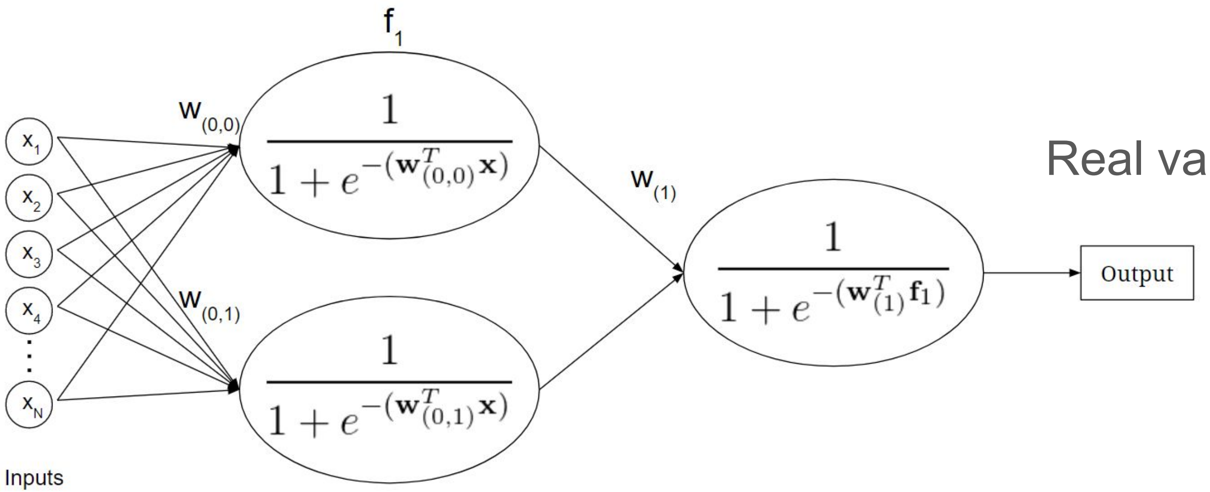
$$E = \frac{1}{2}\sum_{i=1}^{N}(f(x_i) - y_i)^2$$

- Classification (network predicts class probability estimates):

$$E = -\sum_{i=1}^{N}\sum_{j=1}^{d} y_{ij}\log(f(x_{ij}))$$

Gradient Descent

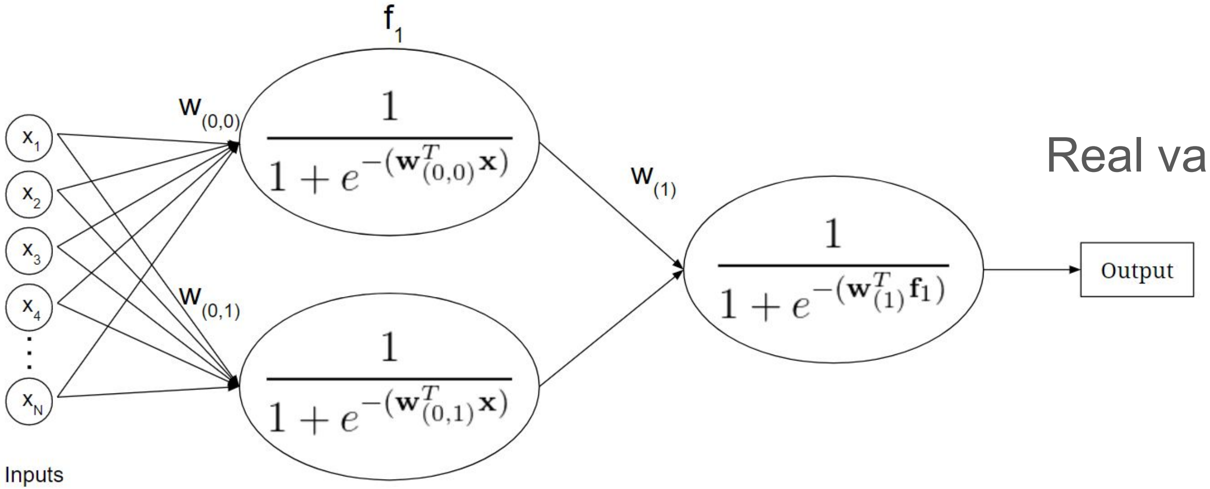
The University of Manchester



- Note the use of the chain rule to compute the derivative

$$\frac{\partial E}{\partial w_{(1)j}} = \frac{\partial E}{\partial f(x)} \frac{\partial f(x)}{\partial (\mathbf{w}_{(1)}^T \mathbf{f}_1)} \frac{\partial (\mathbf{w}_{(1)}^T \mathbf{f}_1)}{\partial w_{(1)j}} = -\sum_i (f(x_i) - y_i) x_{ij}$$

BackProp

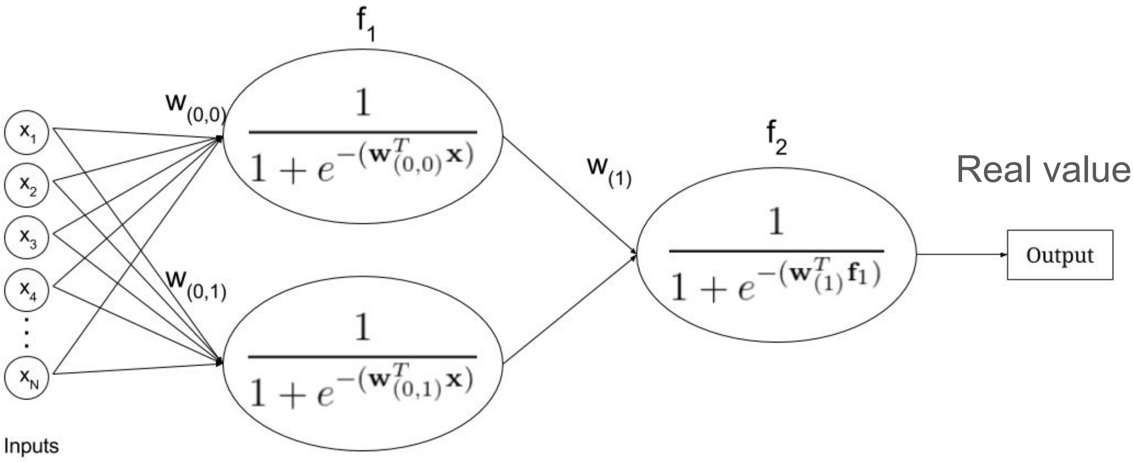


- How do we update $w_{(0,0)}$ and $w_{(0,1)}$?
- We propagate the error through the network.

The University of Manchester

BackProp

Descend to next layer and compute the gradient with respect to $W_{(0,0)}$



$$\frac{\partial E}{\partial w_{(0,0)j}} = \frac{\partial E}{\partial f_2(x)} \frac{\partial f_2(x)}{\partial(\mathbf{w}_{(1)}^T\mathbf{f}_1)} \frac{\partial(\mathbf{w}_{(1)}^T\mathbf{f}_1)}{\partial \mathbf{f}_1} \frac{\partial \mathbf{f}_1(x)}{\partial(\mathbf{w}_{(0,0)}^T\mathbf{x})} \frac{\partial(\mathbf{w}_{(0,0)}^T\mathbf{x})}{\partial w_{(0,0)j}}$$

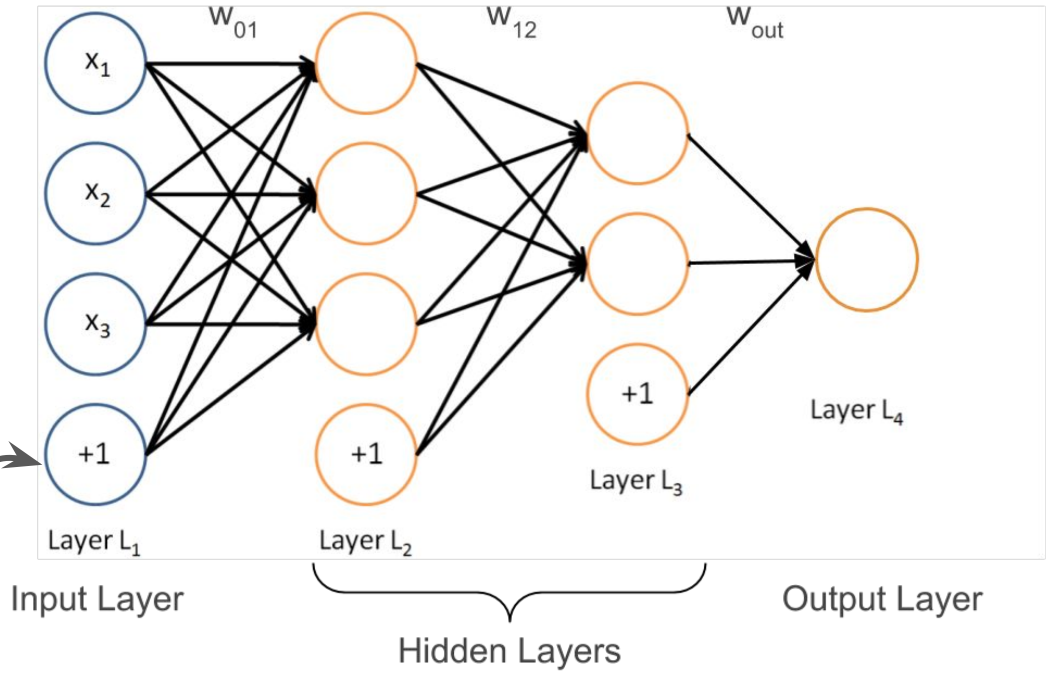
MANCHESTER 1824
The University of Manchester

Deep Neural Network

- Can add more layers and neurons in each layer

- A bias neuron can be used to shift the decision boundary, as in the Perceptron:

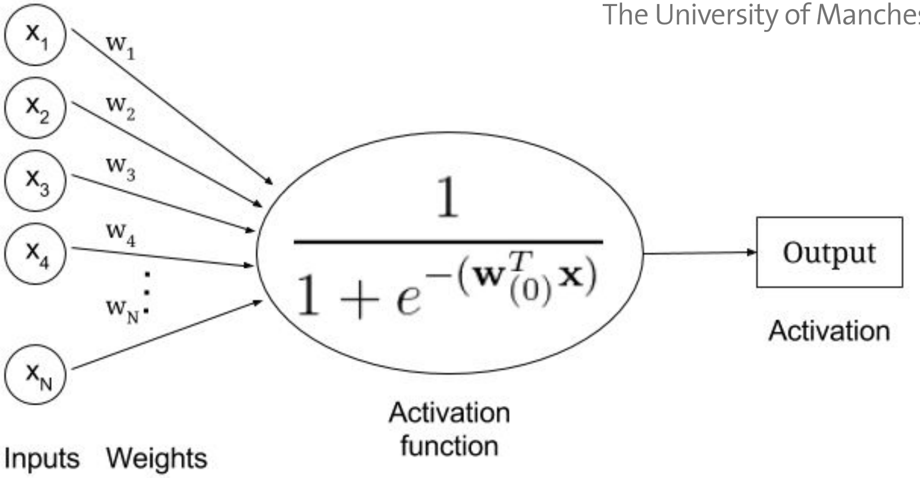
$$f(x) = \mathbf{w}^T\mathbf{x} - \boxed{t}$$

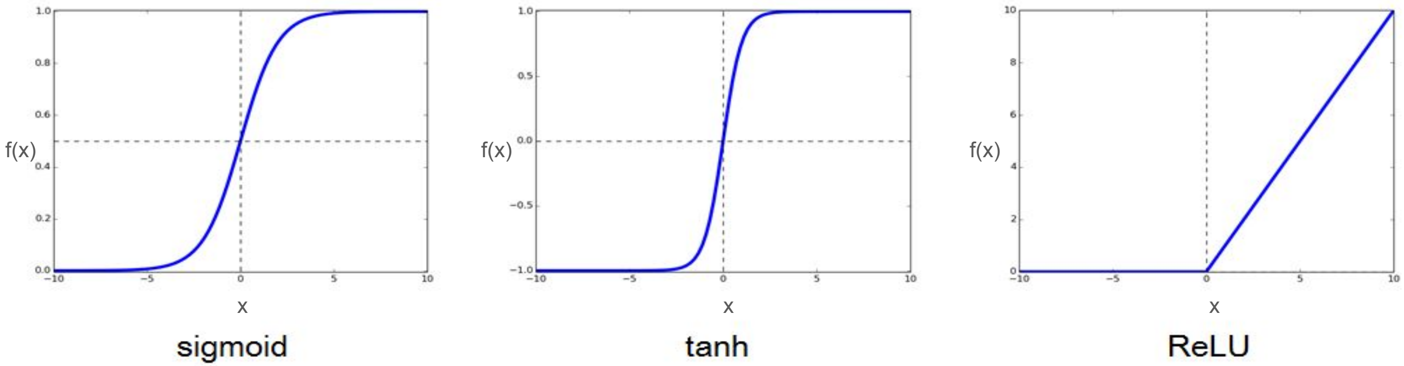


[12]

Activation Functions

- Commonly used functions:





Activation Functions

- Sigmoid
  - Output can be interpreted as probabilities

$$f(x) = \frac{e^x}{1 + e^x}$$

- ReLu (Rectified Linear Unit)
  - No vanishing or exploding gradient

$$f(x) = max(0, x)$$

- Tanh (Hyperbolic Tangent)
  - Converges faster than the sigmoid function

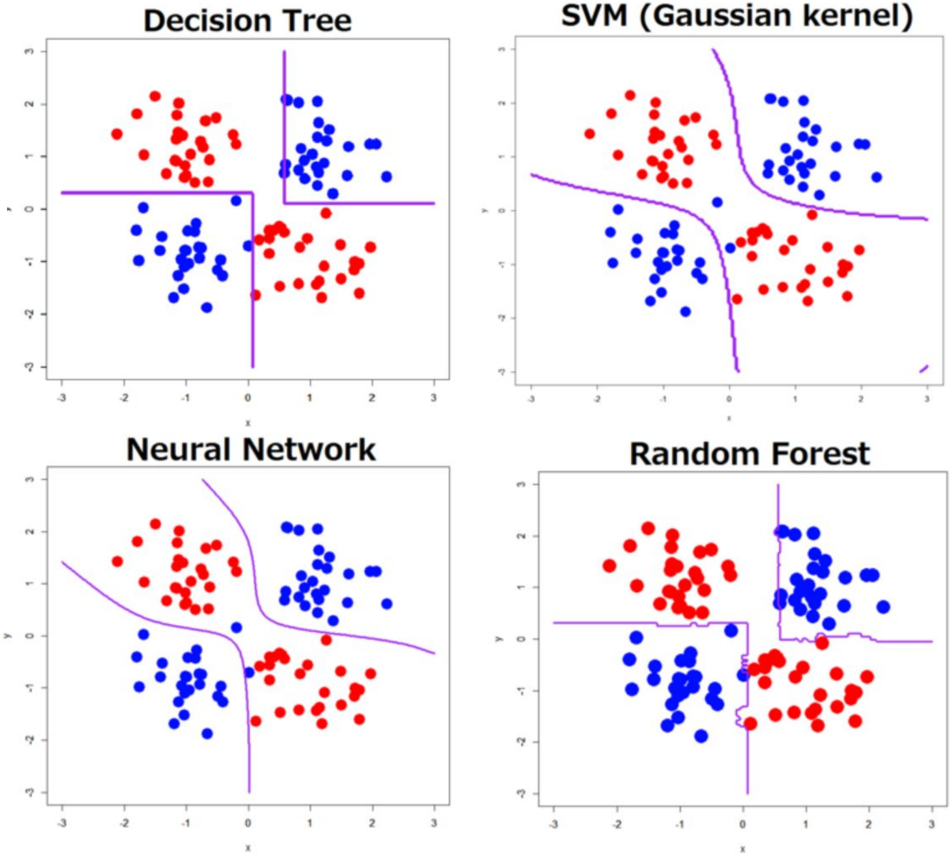
$$f(x) = \frac{1 - e^{-2x}}{1 + e^{-2x}}$$

- SoftMax
  - Generalisation of the logistic function, outputs can be interpreted as probabilities

$$f(x_j) = \frac{e^{x_j}}{\sum_{i=1}^{n} e^{x_i}}$$

Decision boundary

- XOR problem (non-linear)
- Neural Networks are nonlinear models



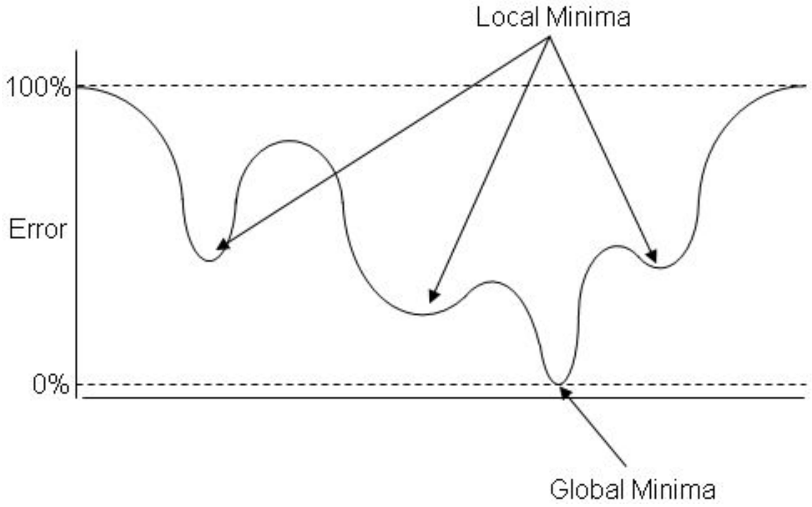
[13]

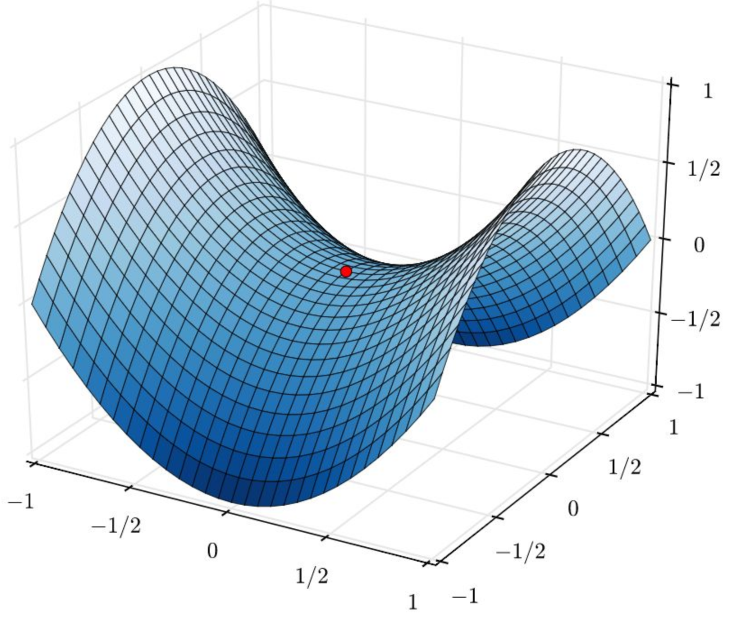
Takasi J. Ozaki, Decision Boundaries for Deep Learning and other Machine Learning classifiers

Potential Issues

Local minima

- Caused by the high dimensional parameter space, which causes points to be saddle points instead





- Because of this, they are not an issue in practice

[14, 15]

Vanishing gradient problem

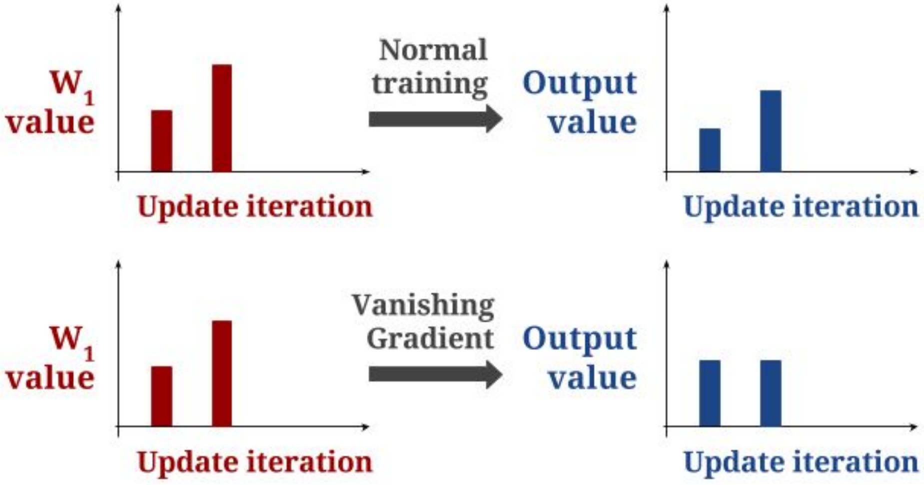
- Appears when a change in a parameter's value causes very small changes in the value of the network output

New parameter value

$$w_1^{t+1} = w_1^t - \alpha \frac{\partial E}{\partial w_1}$$

Old parameter value

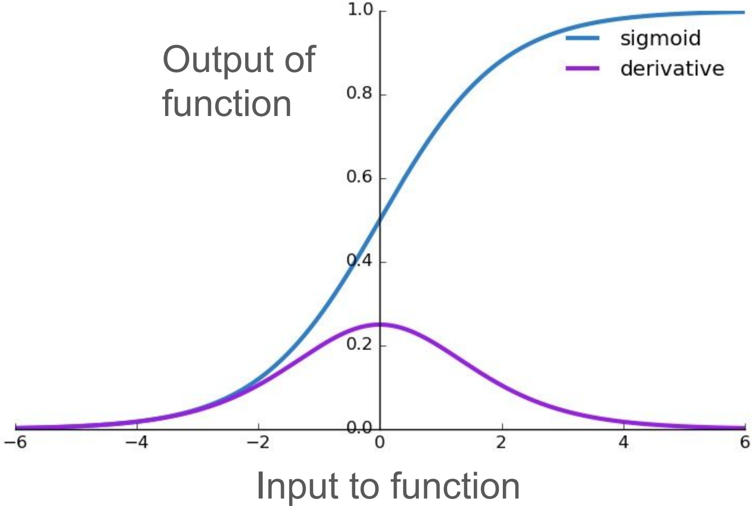
Learning rate



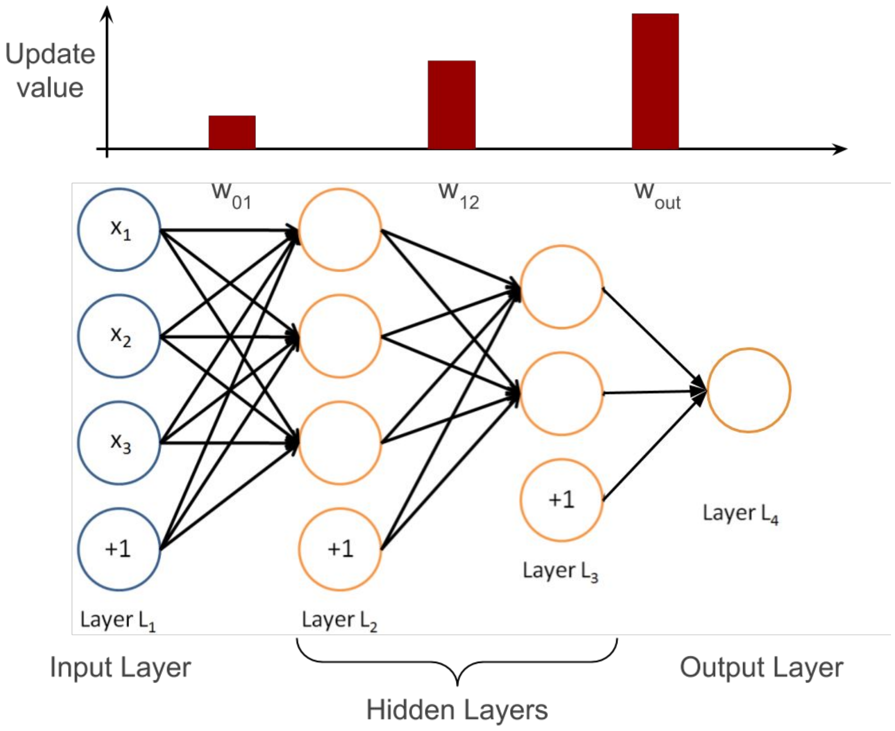
- Manifests in very small gradient values when the update of the parameter is computed

Vanishing gradient problem

- Appears in gradient based methods, caused by some activation functions (sigmoid or tanh)

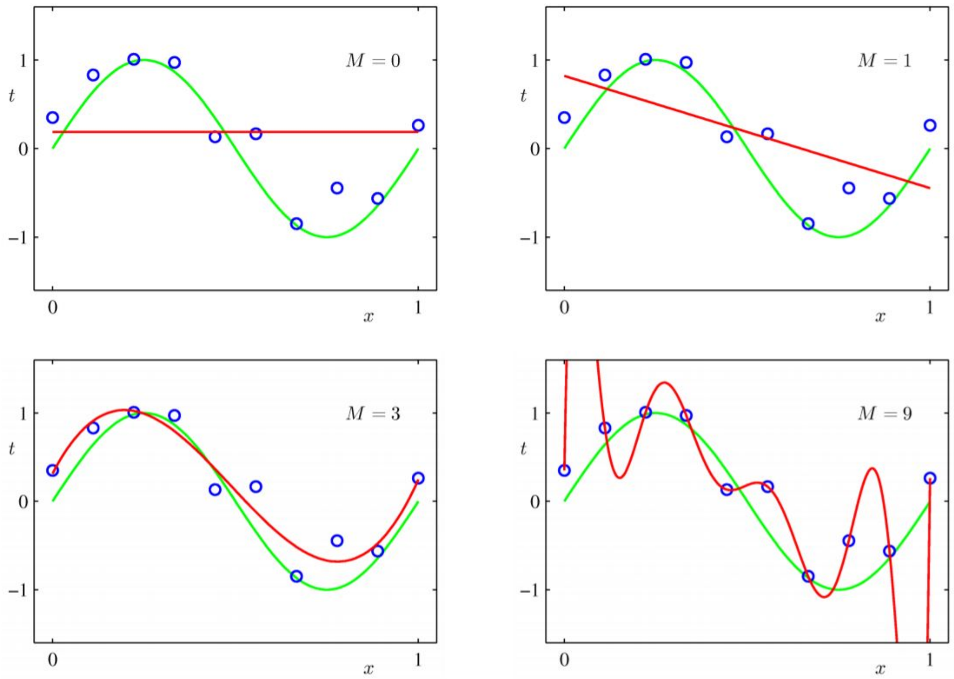


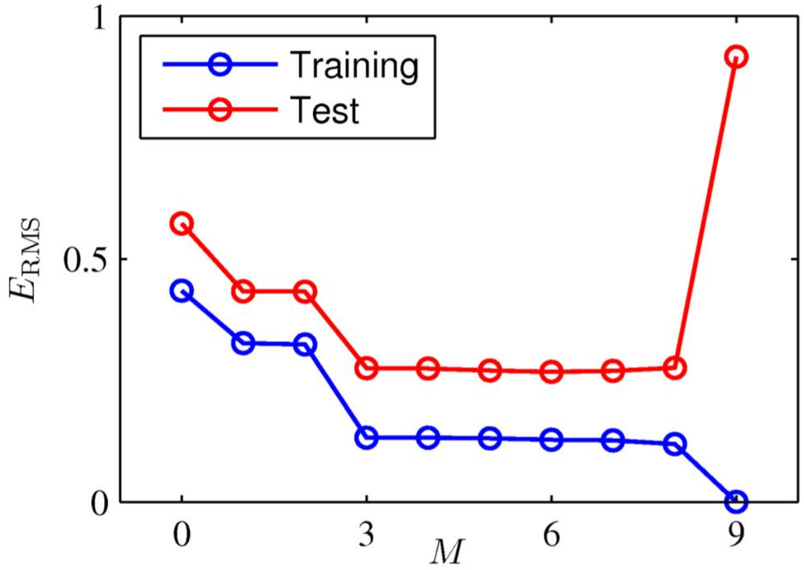
[16, 12]



- Magnified by the addition of hidden layers

Overfitting

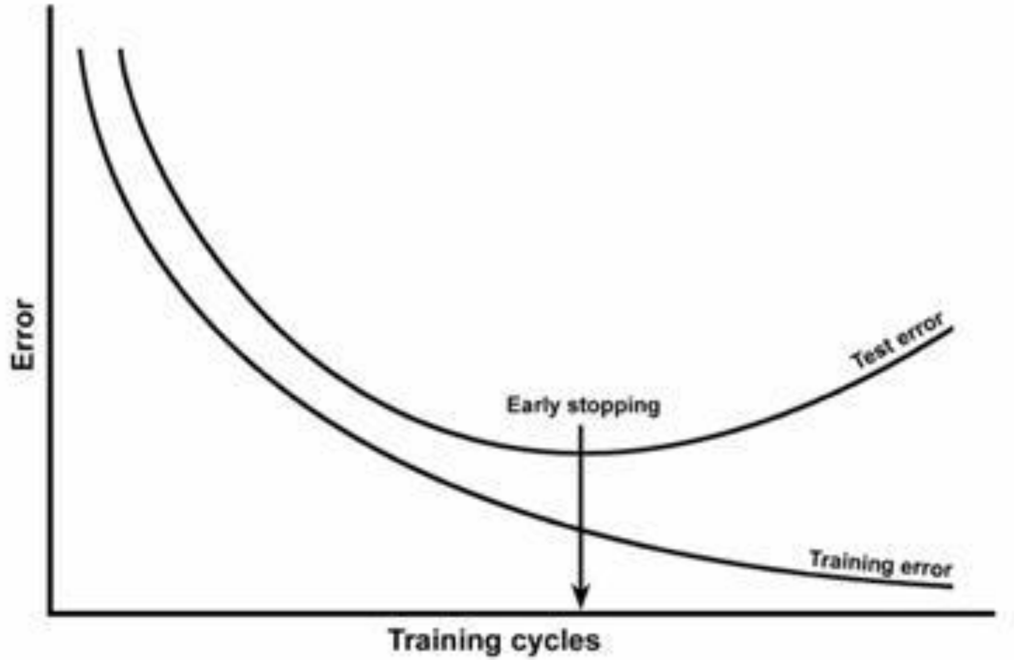




[17]

Preventing Overfitting

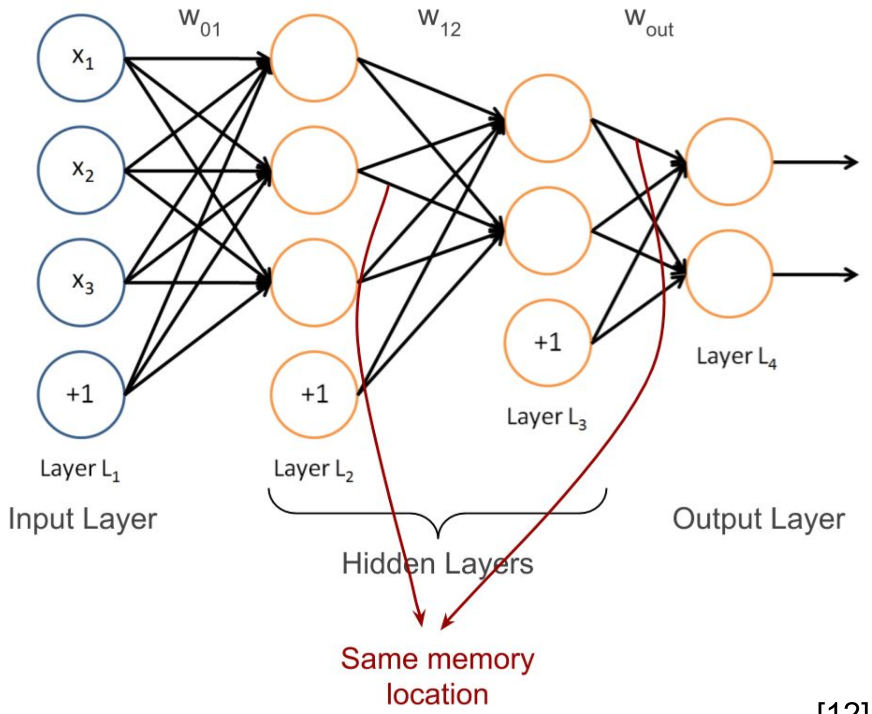
Early Stopping



[18]

Weight sharing

- Parameters are shared by having the values stored in the same memory location
- Decrease amount of parameters at the cost of reducing model complexity
- Mostly used in convolutional and recurrent networks

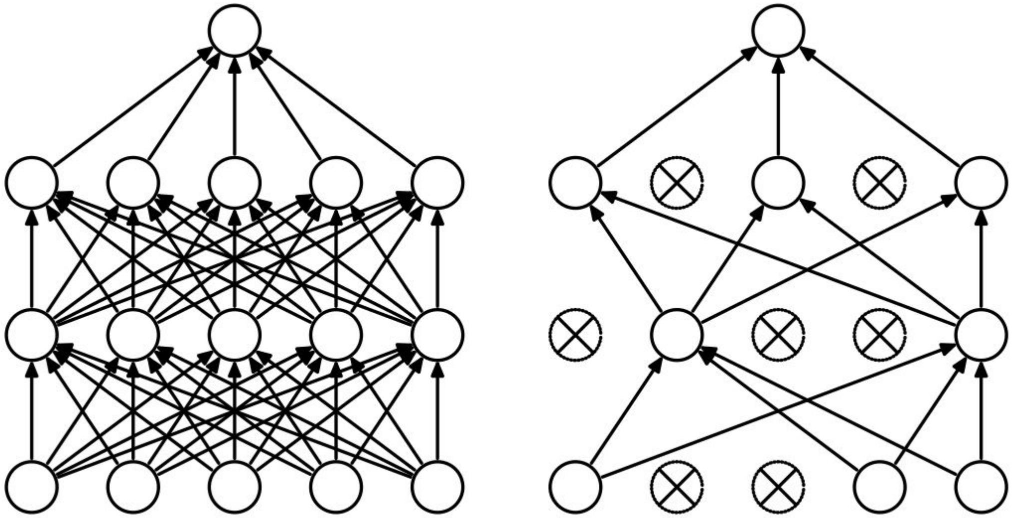


[12]

Dropout

- Randomly omit some units of the network over a training batch (group of training examples)

- Encourage specialization of the generated network to the batch

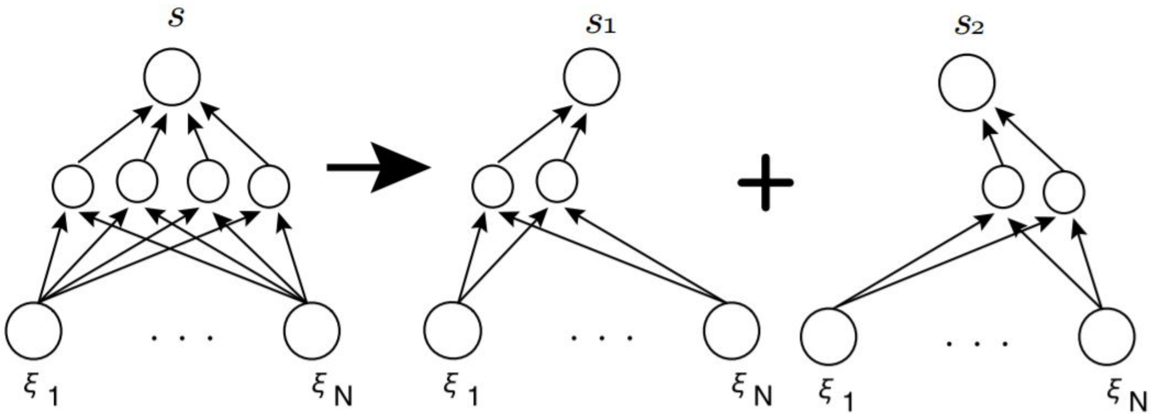


(a) Standard Neural Net (b) After applying dropout.

[19]

Dropout

- It is a form of regularization
- Akin to using an ensemble, each trained on single batches



[20]

Conclusions

Summary

- Impressive performance on difficult tasks has made Deep Learning very popular
- Based on Perceptron and Logistic Regression
- Training is done using Gradient Descent and Backprop
- Error function, activation function and architecture are problem dependent
- Easy to overfit, but there are ways to avoid it

Research Directions

- Understanding more about how Neural Networks learn
- Applications to vision, speech and problem solving
- Improving computational performance, specialised hardware
    - Tensor Processing Units (TPUs)
- Moving towards more biologically inspired neurons
    - Spiking Neurons

Libraries and Resources

- **Tensorflow**: great support and lots of resources
- **Theano**: one of the first deep learning libraries, no multi-GPU support (support discontinued)
- **Keras**: very high level library that work on top of Theano or Tensorflow
- **Lasagne**: similar to Keras, but only compatible with Theano
- **Caffe**: specialised more for computer vision than deep learning
- **Torch**: uses the programming language Lua, has a wrapper for Python

Thank You!

The University of Manchester

References

1. Reynolds E, 2016, *Nvidia unveils deep learning supercomputer*, viewed 10 October 2018, (https://www.wired.co.uk/article/nvidia-supercomputer-deep-learning)
2. Dutt A, 2017, *How is deep learning affecting sports?*, viewed 10 October 2018, (https://yourstory.com/2017/10/how-is-deep-learning-affecting-sports)
3. Burgess M, 2016, *Google's DeepMind wins historic Go contest 4-1*, viewed 10 October 2018, (https://www.wired.co.uk/article/alphago-deepmind-google-wins-lee-sedol)
4. Captain S, 2017, *Intel debuts a deep-learning AI chip to battle Nvidia*, viewed 10 October 2018, (https://www.fastcompany.com/40482484/intel-debuts-a-deep-learning-ai-chip-to-battle-nvidia)
5. Liszewski A, 2017, *Clever camera app uses deep learning to perfectly retouch your photos before you take them*, viewed 10 October 2018, (https://gizmodo.com/clever-camera-app-uses-deep-learning-to-perfectly-retou-1797474282)
6. Geitgey A, 2016, Machine learning is fun part 6: How to do Speech Recognition with Deep Learning, viewed 10 October 2018, (https://medium.com/@ageitgey/machine-learning-is-fun-part-6-how-to-do-speech-recognition-with-deep-learning-28293c162f7a)
7. 2014, Decision support system, viewed 10 October 2018, (http://decision-support-system.blogspot.com/2014/06/decision-support-system.html)
8. 2015,Deep learning experiment, What would a robot see in the mirror?, viewed 10 October 2018, (https://secondrobotics.com/robots/nvidia-jetson-robot-concept/deep-learning-experiments/)
9. 2016, Your personal AI (PAI): Pt 4- Deep Agents (Deep Learning and Natural Intelligence), viewed 10 October 2018, (https://medium.com/@johnsmart/your-personal-sim-pt-4-deep-agents-understanding-natural-intelligence-7040ae074b71)
10. Reinforcement learning, wikipedia, viewed 10 October 2018, (https://en.wikipedia.org/wiki/Reinforcement_learning#/media/File:Reinforcement_learning_diagram.svg)

The University of Manchester

References

11. Brown G, *Linear models*, viewed 12 October 2018, (http://syllabus.cs.manchester.ac.uk/pgt/2019/COMP61011/lectures.php)
12. Varma R, 2016, *Applying neural networks to natural language processing tasks*, viewed 12 October 2018, (https://rohanvarma.me/Neural-NLP/)
13. 2016, *What kind of decision boundaries does deep learning draw?*, viewed 12 October 2018, (https://analyticks.wordpress.com/2016/07/22/what-kind-of-decision-boundaries-does-deep-learning-deep-belief-net-draw-practice-with-r-and-h2o-package/)
14. viewed 12 October 2018, (https://66.media.tumblr.com/515b052849edbd6b6bf35059cc39310a/tumblr_inline_npz2dsoaS81rnd3q0_500.gif)
15. 2017, *Why is Newton's method not more widely used in machine learning?*, viewed 12 October 2018, (https://stats.stackexchange.com/questions/253632/why-is-newtons-method-not-widely-used-in-machine-learning)
16. Varma R, 2018, *Picking loss functions*, viewed 15 October 2018, (https://medium.com/@omkar.nallagoni/activation-functions-with-derivative-and-python-code-sigmoid-vs-tanh-vs-relu-44d23915c1f4)
17. Bishop 2006, Pattern recognition and machine learning
18. Domingues G. et. al., Artificial neural networks on integrated multispectral and SAR data for high-performance predictions of eucalyptus biomass
19. Srivastava N. et. al., Dropout a simple way to prevent neural networks from overfitting, 2014
20. Hara K. et. al., Analysis of dropout learning regarded as ensemble learning, 2017