This summary is based on A 'Brief' History of Neural Nets and Deep Learning by Andrew Kurenkov.

1 A brief history of deep learning

There is a deep learning tsunami over the past several years.

- drastic improvements over reigning approaches towards the hardest problems in AI
- massive investments from industry giants such as Google
- exponential growth in research publications (and ML graduate students)

The birth of machine learning

In 1957, a psychologist Frank Rosenblatt developed perceptron, a mathematical model of neurons in our brain.

A perceptron:

- Takes binary inputs, which are either data or the output of another perceptron (a nearby neuron).
- A special bias input has the value of 1. It allows us to compute more functions using a perceptron.
- Links between neurons are called synapses and the synapses have different strengths.

  Model this by multiplying each input by a continuous valued weight.
- The output depends only on inputs.
- A neuron either fires or not.

The output is 1 if the weighted sum is big enough and the output is 0 otherwise.

Activation function: a non-linear function of the weighted sum. If the weighted sum is above the threshold, the output is 1.

e.g. step function or the sigmoid function.

The perceptron

- based on earlier work by Mcculoch-Pitts
- can represent AND, OR, and NOT (1943).
- big deal: believed that AI is solved if computers could perform formal logical reasoning.
- open question: how do we learn a perceptron model? Rosenblatt answered this question.

Learning a perceptron:

- An idea by Donald Hebb: the brain learns by forming synapses and changing the strenghths of the synapses between the neurons.
- Neuron A repeatedly and persistently takes part in firing neuron B, the brain grows to strength the synapse between A and B.
- For each example, increase the weights if the output is too low and decrease the weights when the output is too high.

A simple algorithm to learn a perceptron:

1. Start with random weights in a perceptron.
2. For a training example, compute the output of the perceptron.

3. If the output does not match the correct input:
4. If the correct output was 0 but the actual output was 1, decrease the weights that had an input of 1.
5. If the correct output was 1 but the actual output was 0, increase the weights that had an input of 1.
6. Repeat steps 2-5 for all the training examples until the perceptron makes no more mistakes

Rosenblatt implemented perceptrons and showed that they can learn to classify simple shapes correctly with 20x20 pixel-like inputs. – Machine learning was born.

The hype around perceptrons

How can we use a perceptron for classification tasks with multiple categories? For example, classify handwritten digits.

- Arrange multiple perceptrons in a layer.
- The perceptrons receive the same inputs.
- Each perceptron learns one output of the function. (Does the inputs belong to a particular class?)

Artificial Neural Networks are simply layers of perceptrons/neurons/units. So far, our network only has one layer - the output layer.

This can be used to classify handwritten digits. Each of the 10 output values represents a digit. The highest weighted sum produces an output of 1 and others produce an output of 0.

The hype around perceptrons:

- Perceptrons are so simple. — basically linear regression. So cannot solve vision or speech recognition problems yet.

- However, a network of such simple units can be powerful and solve complex problems.

In 1958, Rosenblatt said that: Perceptrons might be fired to the planets as mechanical space explorers.

AI winter

The hype irritated other researchers who were skeptical about perceptrons.

- Big problem in AI: formal logical reasoning. teaching computers to manipulate logical symbols using rules.
- In 1969, Marvin Minsky (founder of MIT AI lab) and Seymour Papert (director of the lab) published a book named Perceptrons - a rigorous analysis of the limitations of perceptrons.

  Perceptrons. An Introduction to Computational Geometry. MARVIN MINSKY and SEYMOUR PAPERT. M.I.T. Press, Cambridge, Mass., 1969.
- A notable result: impossible to learn an XOR function using a single perceptron. (XOR function is not linearly separable.).
- They basically said: this approach was a dead end. This publication was believed to lead to the first AI winter - a freeze to funding and publications.

What did Minsky's book show?

We need a multi-layer network to do useful things

- To learn an XOR function or other complex functions.
- Show what a multi-layer neural network looks like.
- Hidden layers are good for feature extraction.

  Facial recognition: The first hidden layer converts raw pixels to the locations of lines, circles, and ovals in the images. The next hidden layer will take the locations of the shapes and determine whether there are faces in the images.

Problem:

- Rosenblatt's learning algorithm does not work for a multi-layer neural network.
- How do we adjust the weights in the middle layers? Use backpropagation.

Backpropagation for neural nets

- Calculate the errors in the output layer.
- Propagate these errors backwards to the previous hidden layer. Blame the previous layer for some of the errors made.
- Propagate these errors backwards again to the previous hidden layer.
- When we change weights in any layer, the errors in the output layer changes.
- Use an optimization algorithm to find weights that minimize the errors.

History of backpropagation:

- Derived by multiple researchers in the 60s
- Implemented by Seppo Linnainmaa in 1970.
- Paul Werbos first proposed that backpropagation can be used for neural nets. (analyzed it in depth in his 1974 PhD thesis).

The atmosphere at the time during the AI winter:

- In his 1969 book, Minsky showed that: we need to use multi-layer perceptrons even to represent simple nonlinear functions such as the XOR mapping.
- Many researchers had tried and failed to find good ways to train multi-layer perceptrons.

- Neural networks were an dead end. Lack of academic interest in this topic. The community lost faith.
- Paul Werbos did not publish his idea until 1982.

The rise of back propagation

More than a decade after Werbos' thesis, the idea of back propagation was finally popularized.

Rumelhart, D. E., Hinton, G. E., and Williams, R. J. (1986). Learning representations by back-propagating errors. Nature, 323, 533–536.

- Idea was rediscovered multiple times before this paper
- The paper stands out for concisely and clearly stating the idea.
- Finally succeeded in making this idea well-known.
- Paper is identical to how the concept is explained in textbooks and AI classes.
- Wrote another in-depth paper to specifically address the problem pointed out by Minsky.

Neural networks are back. We know how to train multi-layer neural networks to solve complex problems.

Neural networks are back

We know how to train multi-layer neural networks. Become super popular. The ambitions of Rosenblatt seems to be within reach.

1989 a key finding: A mathematical proof that multi-layer neural networks can implement any function.

Kurt Hornik, Maxwell Stinchcombe, Halbert White, Multilayer feedforward networks are universal approximators, Neural Networks, Volume 2, Issue 5, 1989,

Pages 359-366, ISSN 0893-6080, http://dx.doi.org/10.1016/0893-6080(89)90020-8.

In 1989 a significant real-world application of backpropagation: handwritten zip code recognition.

- US postal service was desperate to be able to sort mails automatically. Recognizing messy handwriting was a major challenge.
- Yann LeCun and others at the AT&T Bell Labs
- LeCun, Y; Boser, B; Denker, J; Henderson, D; Howard, R; Hubbard, W; Jackel, L, "Backpropagation Applied to Handwritten Zip Code Recognition," in Neural Computation , vol.1, no.4, pp.541-551, Dec. 1989 89
- The method was later used for a nationally deployed cheque reading system in the mid 90. (Show video.) (At some point in the late 1990s, one of these systems was reading 10 to 20% of all the checks in the US.)
- Highlighted a key modification of neural networks towards modern machine learning: extracting local features and combining them to form higher order features

Convolutional neural networks:

- Each neuron extracts a local feature anywhere inside an image. Force a hidden unit to only combine local sources of information.

  Used to be that each neuron is passed the entire image. Now each neuron is only passed a portion of the image and it's looking for a local feature in that portion.
- Passing a filter through an image and only picks up whether the image has this feature (a horizontal line, a 45 degree line, etc). A magnifying glass sliding across the image. Capable of only recognizing one thing. Records the location of each match.
- The following layers combine local features to form higher order features.

Why is this a good idea?

- Without specialized neurons, need to learn the same feature many times in different parts of the image.
- Only one neuron with fewer weights can learn the feature much faster.

A new winter dawns

Researchers started using neural networks for many applications

- Image compression
- Learning a probability distribution
- Playing games such as Backgammon, Chess, and Go.
- Speech recognition

Neural networks with many layers trained with backpropagation did not work well.

- Not as well as simpler networks.
- Backpropagation relies on finding the error at the output layer and successively splitting up blame for it for prior layers.
- With many layers this splitting of blame ends up with either huge or tiny numbers and the resulting neural net just does not work very well.

In mid 90s, a new AI winter began.

- General perception that neural networks do not work well.
- Computers were not fast enough. The algorithms were not smart enough. People were not happy.

- In contrast, support vector machine (similar to a 2-layer neural network) was shown to work better than neural networks. (LeCun showed this for handwritten digit recognition.)
- A random forest (multiple decision trees) also worked very well.

A dark time for neural nets (early 2000s):

- LeCun and Hinton's papers were routinely rejected from conferences due to their subjects being neural nets.
- Hinton found an ally: the Canadian government.
- The Canadian Institute for Advanced Research (CIFAR) encouraged basic research without application.
- Hinton secured funding from CIFAR and moved to Canada in 1987.
- With modest funding, they continued working.

A conspiracy was hatched

- Rebranded the neural nets with the term "deep learning".
- A significant breakthrough to rekindle interest in neural networks and started the deep learning movement.

  Hinton, G. E., Osindero, S., & Teh, Y. W. (2006). A fast learning algorithm for deep belief nets. Neural computation, 18(7), 1527-1554.
- Key idea: Neural nets with many layers could be trained well if the initial weights are chosen in a clever way rather than randomly.

  (Set the initial weights by training each layer separately using unsupervised learning.)
- Proof: Hinton and his students applied deep learning to speech recognition. On a standard speech recognition dataset, the best performance was

achieve a decade ago. They improved on this performance record, which was impressive.

Mohamed, A. R., Sainath, T. N., Dahl, G., Ramabhadran, B., Hinton, G. E., & Picheny, M. (2011, May). Deep belief networks using discriminative features for phone recognition. In Acoustics, Speech and Signal Processing (ICASSP), 2011 IEEE International Conference on (pp. 5060-5063). IEEE.

The importance of brute force

The benefit of fast parallel computation and lots of training data

- CPUs hit a celing (can be run in weak parallelism). High-end graphics cards were powerful and can be used in parallel.
- Collect lots of training data – can prevent overfitting. Neural networks are extremely complex. Used on a small data set, it is easy to overfit.

Students of Hinton Dahl and Mohamed went to Microsoft and worked on deep learning there.

Another student of Hinton Jaitly went to work at Google. Google soon used deep learning to power Android's speech recognition.

Andrew Ng and Jeff Dean formed Google brain. They built even bigger neural nets (1 billion weights) to train deep belief nets to recognize the most common objects in YouTube videos. The net discovered cats (also good at detecting human faces and human body parts).

Why does backpropagation not work well?

Why the old approaches did not work well?

Glorot, X., & Bengio, Y. (2010). Understanding the difficulty of training deep feedforward neural networks. In International conference on artificial intelligence and statistics (pp. 249-256).

- The default activation function was not a good choice.

  The non-differentiable function and very simple function f(x)=max(0,x) (rectified linear unit) is the best.

- Weights should be chosen in different scales depending on the layers they are in.

In 2012, Hinton entered the ImageNet competition using deep convolutional neural networks. They did far better than the next closest entry. Their error rate was 15.3%, whereas the second closest was 26.2%. CNN gained respect from the vision community.

The deep learning tsunami began here in 2012. it has been growing and intensifying to this day. No winter is in sight.

2 Summary

1. (Frank Rosenblatt 1957) A perceptron can be used to represent and learn logic operators (AND, OR, and NOT). At the time, it was widely believed that AI is solved if computers can perform formal logical reasoning.

2. (Marvin Minsky 1969) published a rigorous analysis of the limitations of perceptrons. Minsky made two claims: We need to use multi-layer perceptrons to represent simple non-linear functions such as XOR. No one knows a good way to train multi-layer perceptrons. This led to the first AI winter (70s to 80s).

3. In 1986, Rumelhart, Hinton and Williams in their Nature article precisely and concisely explained how we can use the back-propagation algorithm to train multi-layer perceptrons.

4. In mid 90s, people found that multi-layer neural networks trained with back-propagation don't work well compared to simpler models (such as support vector machines and random forests). This led to the second AI Winter around mid 90s.

5. In 2006, Hinton, Osindero and Teh showed that backpropagation works if the initial weights are chosen in a smart way. This brought neural networks back into the limelight. In 2012, Hinton entered the ImageNet competition using deep convolutional neural networks and did far better than the next closest entry (their error rate was 15.3% rather than 26.2%). The deep learning tsunami continues today.

Lessons learned summarized by Geoffrey Hinton:

- Our labeled datasets were thousands of times too small.

  Complex neural networks are prone to overfitting. We need lots of training data to prevent overfitting.

- Our computers were millions of times too slow.

  We need a lot of computational power to train massive neural networks.

- We initialized the weights in a stupid way.
  Weights should be chosen in a clever way rather than randomly.
- We used the wrong type of non-linearity.
  We should have used a different activation function.