Deep Learning: Lecture 0

Keith L. Downing

The Norwegian University of Science and Technology (NTNU)
Trondheim, Norway
keithd@idi.ntnu.no

January 9, 2024
Representations for AI Problem Solving

The Big Choice

1. **Crafted by Human Engineers**: Hard work for knowledge engineers, but the AI’s decision making is normally straightforward to understand.

2. **Formed by the AI System**: Easy to setup (e.g. via SciKit Learn) but VERY difficult to interpret results, particularly with non-symbolic systems (e.g. neural nets).
Outline

1. A Brief History of Neural Networks
2. Linear Separability
3. Brief Overview of Backpropagation
4. A Few Example Applications
The Early History of Neural Networks

- McCulloch + Pitts (1943) - neuron model similar to logic gates: no weights and no learning, but special excitatory and inhibitory connections.
- Rosenblatt (1958) - The perceptron, a 3-layered network. Today, we call his output layer a perceptron, since connections between other two layers were not adaptive.
- Widrow + Hoff (1960) - Adalines + Delta Rule for training them, where error signal is based on the weighted sum of inputs, not the output of an activation function.
- Minsky + Papert (1969) - Proved that non-linearly-separable functions (e.g. XOR) could not be represented by a two-layered neural network (regardless of the type of neuron). Since a) Most hard data sets are not linearly separable, and b) Delta rule fails for nets with more than 2 layers → Nets for hard data sets cannot be trained!
The Ressurection of Neural Networks


- **Hibernation (1995-2005)** - Trappings of local minima and failure of deep nets (due to attenuated backpropagation signals, i.e. gradients) became glaring weaknesses that prevented scaling up.

- **The Deep Learning Revolution (2006-present)** - Unsupervised pre-training (later found unnecessary) + many small (but significant) changes/extensions to backpropagation + major hardware improvements + **BIG DATA** → Learning in nets with 100+ layers!!
The Perceptron

Keith L. Downing
Deep Learning: Lecture 0
The Delta Rule

$T = \text{target output value}$

$\delta = \text{error}$

$\Delta w = \eta \delta X$

$\Delta w = \eta (T - Y)$
XOR: The (Near) Death of Neural Networks

- (0.5)x + (0.5)y ≥ 1
  - y ≥ 2 - x
- (0.5)x + (0.5)y ≥ 0
  - y ≥ -x
- (??)x + (??)y ≥ ??
- y ≥ 2 - x

Keith L. Downing
Deep Learning: Lecture 0
Linear Separability of Data

\[ w_x x + w_y y \geq t_z \]

Activation function:

\[ \text{activation}_z \]

Sample data points:

\[ X, Y \]

\[ x, y \]

\[ t_z = 1 \]

Keith L. Downing

Deep Learning: Lecture 0
Adding a Hidden Layer

Not linearly separable → Need hidden layer with non-linear act func.

\[ u = 1 \]
\[ v = 1 \]
\[ z = 0 \]

\[ y = x + 2 \]
\[ y = x - 2 \]
Both operations – (a) summing weighted inputs (including biases), and (b) firing activation functions – move points around in multi-dimensional spaces, with one space per layer.

# dimensions of a space = # neurons in that layer.
Matrix multiplication of a vector by a weight matrix (plus addition of biases) = affine transformation.

These preserve colinearity among (triples of) points, but non-linear transformations (in some activation functions) do not.

To transform points to linearly separable, requires at least one non-linear activation layer.
Backpropagation

Training/Test Cases: \{(d_1, r_1) (d_2, r_2) (d_3, r_3)\ldots\}

\[ E = r_3 - r^* \]

\[ \frac{dE}{dW} \]

Training

Test

N times, with learning

1 time, without learning

Keith L. Downing
Deep Learning: Lecture 0
Even massive neural nets with diverse layer types use backprop.

Transformers and Large Language Models

Standard Transformers have 2 copies of the weight matrix and RELU, in series, in both encoder and decoder modules.
IBM’s DECTalk: **several man years** of work → Reading Machine.

**NETtalk:** 10 **hours** of backprop training on 1000-word text (T1000).

- 95% accuracy on T1000; 78% accuracy on novel text.
- Improvement during training sounds like a child learning to read.
- Concept layer is key: 79 different (overlapping) clouds of active neurons gradually form, with each mapping to one of the 79 phonemes.
Endless Applications of Neural Networks

- Stock and commodity price predictions
- Electrical load predictions for the power industry
- Detection of disease from MRI images
- Facial recognition
- Colorization of old black-white movies.
- Natural language interpretation, generation and translation.
- Production of captions for images.
- Control of self-driving vehicles.
- Automated journalism: given data (sports, financial, etc), write article.
- Generation of art, poetry, music and (more recently) long prose and images.
- Generation and analysis of complex text.
- Commonsense problem solving
- Writing computer code
Me

I have two rubber bands, a bag of malted milk balls, two pencils and a hungry crow. In 50 words or less, explain how to make a game out of these things.

... 2 seconds later...

ChatGPT 3.5

Crow Treat Toss: Stretch rubber bands between pencils to create slingshots. Launch malted milk balls into the air for the hungry crow to catch mid-flight. Award points for successful catches. Enjoy a playful feeding game with the clever crow, fostering interaction and entertainment.
Big Data - *Data is the new oil*

GPUs - greatly speed up the complex calculations of backpropagation.

Convolution nets - based on mammalian visual processing.

LSTMs - slick implementation of recurrence adds critical memory of varying durations.

Dropout - deactivation of random subsets of neurons improves generalization.

Rectified Linear Units (ReLU) - very simple activation function reduces the vanishing-gradient problem → backprop works in very deep networks.

Attention and Transformers - Special vector representation that helps contextualize information within a sequence + a recurrence-free, multi-layered module for processing it.
“Everybody loves somebody sometime”

\( \forall X \text{ Person}(X) \Rightarrow \exists Y, T : \text{Person}(Y) \land \text{Time}(T) \land \text{Loves}(X, Y, T) \)

“My wife seems to trust Roomba more than me to sweep the house”

\( \text{Has}_\text{Task}(\text{Jane-Doe, Sweep(House)}) \land \text{Married}(\text{Jane-Doe, Keith}) \land \exists X : \text{Helps}(X, \text{Jane-Doe, Sweep(House)}) \Rightarrow [\text{prob}(X = \text{Roomba}) = 0.75] \)

Chipped Beef on Toast (a.k.a. $\%\%\%$ on a shingle):

This was a “Dad special” that we suffered through on Sunday mornings. He learned the recipe while serving in the Army reserves in Texas. It was shown being served to troops in a 2023 WW-II series on Netflix. It resembles fish scales in cream, and tastes even worse.
Read about *hat trick* $\rightarrow$ embeddings for *hat* and *three* become more similar.
The Universe of Deep Learning

General DL
Feed Fwd + Backprop over many layers

Sequence Models
Recurrent Nets, LSTM, GRU

Image Models
Convolution Nets (CNNs)

Others
Deep Reinforcement Learning
Adversarial Nets
Unsupervised Nets

* Nuts and Bolts of Applying Deep Learning (Andrew Ng, 2016, YouTube)