Neural Networks

Questions

Motivation

Applications

Introduction

Neural Networks - Questions

- What tasks are machines good at doing that humans are not?
- What tasks are humans good at doing that machines are not?
- What tasks are both good at?
- What does it mean to learn?
- How is learning related to intelligence?
- What does it mean to be intelligent? Do you believe a machine will ever be built that exhibits intelligence?
- Have the above definitions changed over time?
- If a computer were intelligent, how would you know?
- What does it mean to be conscious?
- Can one be intelligent and not conscious or vice versa?

Neural Networks - Motivation for neural neworks

- Scientists are challenged to use machines more effectively for tasks currently solved by humans.
- Symbolic Rules do not always reflect processes actually used by humans
- Traditional computing excels in many areas, but not in others.

Neural Networks - Types of Applications

Machine learning: Having a computer program itself from a set of examples so you do not have to program it yourself is a strong focus of neural networks that learn from a set of examples.

Optimization: given a set of constraints and a cost function, how do you find an optimal solution? E.g. traveling salesman problem.

Classification: grouping patterns into classes: i.e. handwritten characters into letters.

Associative memory: recalling a memory based on a partial match.

Regression: function mapping

Neural Networks - Types of Applications (cont.)

Cognitive science:

- Modeling higher level reasoning: language, problem solving
- Modeling lower level reasoning: vision, speech recognition, speech generation

Neurobiology: Modeling models of how the brain works.

- neuron-level
- higher levels: vision, hearing, etc., overlap with cognitive science.

Mathematics: Nonparametric statistical analysis and regression.

Philosophy: Can human souls/behavior be explained in terms of symbols, or does it require something lower level, like a neurally based model?

Neural Networks - Where are neural networks employed?

Signal processing: suppress line noise, with adaptive echo canceling, blind source separation

Control: e.g. backing up a truck: cab position, rear position, and match with the dock get converted to steering instructions. Manufacturing plants for controlling automated machines.

Siemens successfully uses neural networks for process automation in basic industries, e.g., in rolling mill control more than 100 neural networks do their job, 24 hours a day

Robotics - navigation, vision recognition

Pattern recognition, i.e. recognizing handwritten characters, e.g. the current version of Apple's Newton uses a neural net

Medicine, i.e. storing medical records based on case information

Neural Networks - Where are neural networks employed? (cont)

Speech production: reading text aloud (NETtalk)

Speech recognition

Vision: face recognition , edge detection, visual search engines

Business,e.g.. rules for mortgage decisions are extracted from past decisions made by experienced evaluators, resulting in a network that has a high level of agreement with human experts.

Financial Applications: time series analysis, stock market prediction

Data Compression: speech signal, image, e.g. faces

Game Playing: backgammon, chess, go, ...

Neural Networks - Applications for Neural Networks

Neural networks are applicable in virtually every situation in which a relationship between the predictor variables (independents, inputs) and predicted variables (dependents, outputs) exists, even when that relationship is very complex and not easy to articulate in the usual terms of "correlations" or "differences between groups." A few representative examples of problems to which neural network analysis has been applied successfully are:

Engine management**.** Neural networks have been used to analyze the input of sensors from an engine. The neural network controls the various parameters within which the engine functions, in order to achieve a particular goal, such as minimizing fuel consumption.

Neural Networks - Applications for Neural Networks (cont)

Detection of medical phenomena**.** A variety of health-related indices (e.g., a combination of heart rate, levels of various substances in the blood, respiration rate) can be monitored. The onset of a particular medical condition could be associated with a very complex (e.g., nonlinear and interactive) combination of changes on a subset of the variables being monitored. Neural networks have been used to recognize this predictive pattern so that the appropriate treatment can be prescribed.

- Stock market prediction**.** Fluctuations of stock prices and stock indices are another example of a complex, multidimensional, but in some circumstances at least partially-deterministic phenomenon. Neural networks are being used by many technical analysts to make predictions about stock prices based upon a large number of factors such as past performance of other stocks and various economic indicators.

Neural Networks - Applications for Neural Networks (cont)

Credit assignment**.** A variety of pieces of information are usually known about an applicant for a loan. For instance, the applicant's age, education, occupation, and many other facts may be available. After training a neural network on historical data, neural network analysis can identify the most relevant characteristics and use those to classify applicants as good or bad credit risks.

Monitoring the condition of machinery**.** Neural networks can be instrumental in cutting costs by bringing additional expertise to scheduling the preventive maintenance of machines. A neural network can be trained to distinguish between the sounds a machine makes when it is running normally ("false alarms") versus when it is on the verge of a problem. After this training period, the expertise of the network can be used to warn a technician of an upcoming breakdown, before it occurs and causes costly unforeseen "downtime."

Neural Networks - learning algorithms - what is machine learning?

It is very hard to write programs that solve problems like recognizing a face.

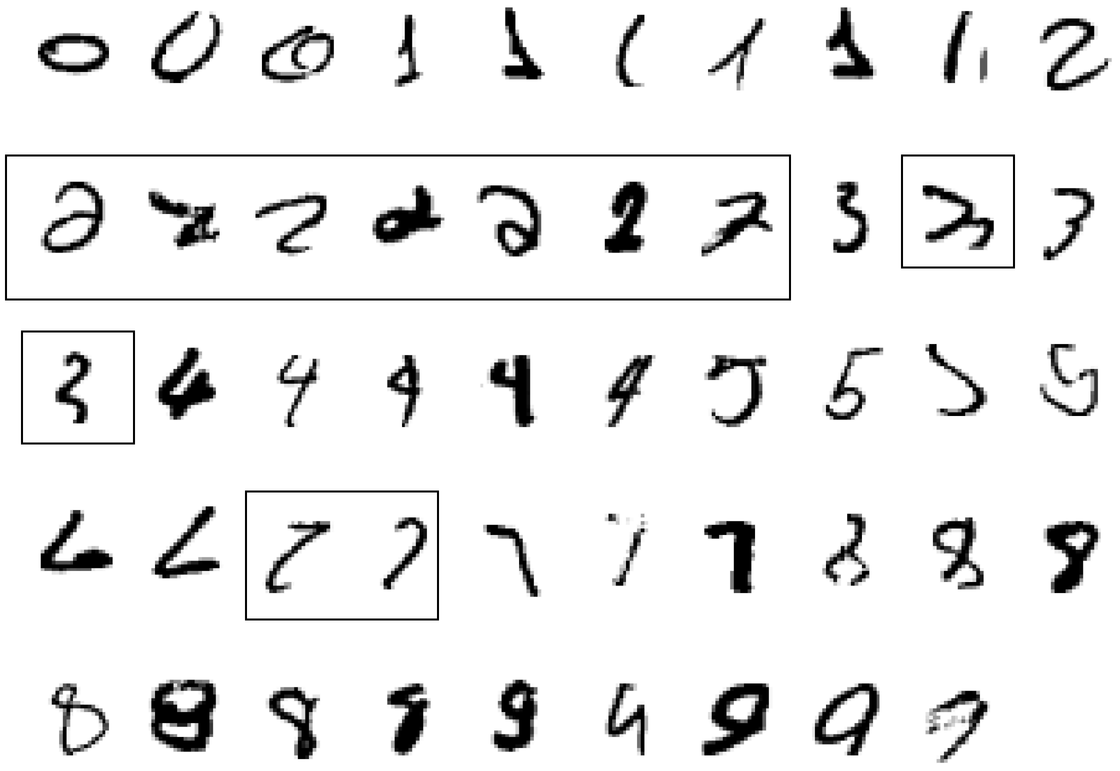
- We don't know what program to write because we don't know how its done.
- Even if we had a good idea about how to do it, the program might be horrendously complicated.

Instead of writing a program by hand, we collect lots of examples that specify the correct output for a given input.

A machine learning algorithm then takes these examples and produces a program that does the job.

- The program produced by the learning algorithm may look very different from a typical hand-written program. It may contain millions of numbers.
- If we do it right, the program works for new cases as well as the ones we trained it on.

Neural Networks - It is very difficult to say what makes a 2



Neural Nets - Some examples of tasks that are best solved by using a learning algorithm

Recognizing patterns:
- Facial identities or facial expressions
- Handwritten or spoken words

Recognizing anomalies:
- Unusual sequences of credit card transactions
- Unusual patterns of sensor readings in a nuclear power plant

Prediction:
- Future stock prices
- Future currency exchange rates

Neural Nets - The goals of neural computation

To understand how the brain actually works

- Its very big and very complicated and made of yukky stuff that dies when you poke it around

To understand a new style of computation

- Inspired by neurons and their adaptive connections
- Very different style from sequential computation
  - should be good for things that brains are good at (e.g. vision)
  - Should be bad for things that brains are bad at (e.g. 23 x 71)

To solve practical problems by developing novel learning algorithms

- Learning algorithms can be very useful even if they have nothing to do with how the brain works

Neural Nets - A typical cortical neuron

Gross physical structure:

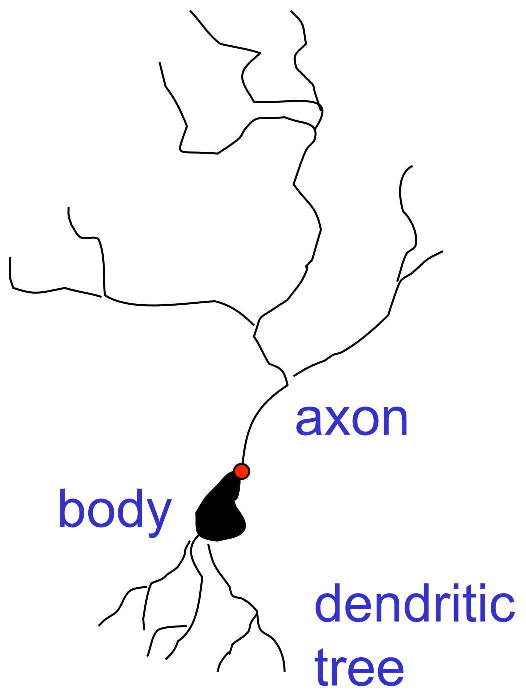
- There is one axon that branches
- There is a dendritic tree that collects input from other neurons

Axons typically contact dendritic trees at synapses

- A spike of activity in the axon causes charge to be injected into the post-synaptic neuron

Spike generation:

- There is an axon hillock that generates outgoing spikes whenever enough charge has flowed in at synapses to depolarize the cell membrane



Neural Networks - Synapses

When a spike travels along an axon and arrives at a synapse it causes vesicles of transmitter chemical to be released

- There are several kinds of transmitter

The transmitter molecules diffuse across the synaptic cleft and bind to receptor molecules in the membrane of the post-synaptic neuron thus changing their shape.

- This opens up holes that allow specific ions in or out.

The effectiveness of the synapse can be changed

- vary the number of vesicles of transmitter
- vary the number of receptor molecules.

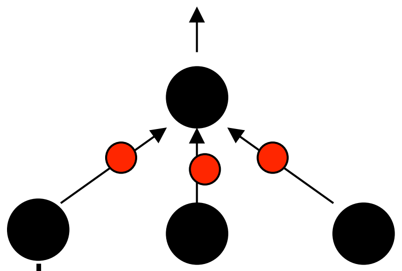
Synapses are slow, but they have advantages over RAM

- Very small
- They adapt using locally available signals (but how?)

Neural Networks - How the brain works

Each neuron receives inputs from other neurons

- Some neurons also connect to receptors
- Cortical neurons use spikes to communicate
- The timing of spikes is important



The effect of each input line on the neuron is controlled by a synaptic weight

- The weights can be positive or negative

The synaptic weights adapt so that the whole network learns to perform useful computations

- Recognizing objects, understanding language, making plans, controlling the body

•You have about 10^{11} neurons each with about 10^{3} weights

- A huge number of weights can affect the computation in a very short time. Much better bandwidth than a computer.

Neural Networks - Modularity and the brain

Different bits of the cortex do different things.

- Local damage to the brain has specific effects
- Specific tasks increase the blood flow to specific regions.

But cortex looks pretty much the same all over.

- Early brain damage makes functions relocate

Cortex is made of general purpose stuff that has the ability to turn into special purpose hardware in response to experience.

- This gives rapid parallel computation plus flexibility.
- Conventional computers get flexibility by having stored programs, but this requires very fast central processors that perform large computations sequentially.

Neural Networks - Idealized neurons

To model things we have to idealize them (e.g. atoms)

- Idealization removes complicated details that are not essential for understanding the main principles
- Allows us to apply mathematics and to make analogies to other, familiar systems.
- Once we understand the basic principles, its easy to add complexity to make the model more faithful

It is often worth understanding models that are known to be wrong (but we mustn't forget that they are wrong!)

- E.g. neurons that communicate real values rather than discrete spikes of activity.

Neural Networks - Linear neurons

- These are simple but computationally limited
  - If we can make them learn we may get insight into more complicated neurons

bias

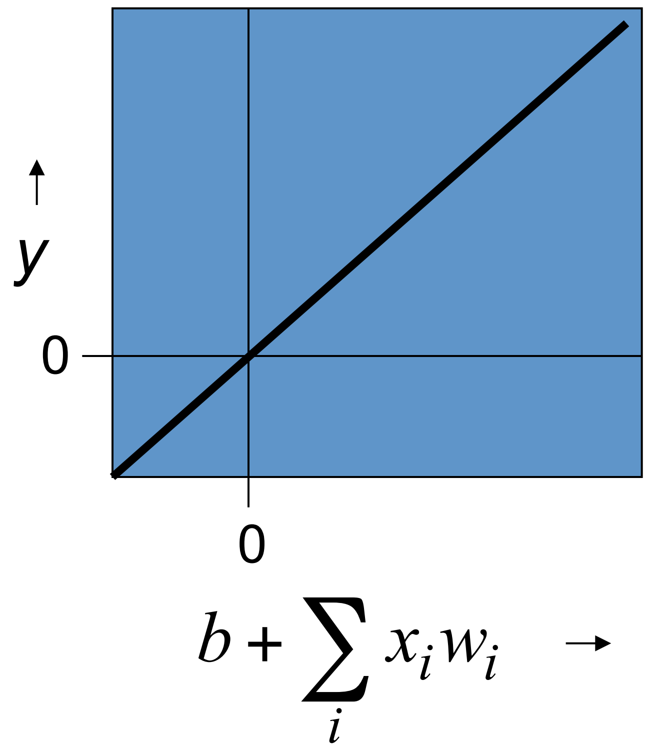
ith input

$$y = b + \sum_i x_i w_i$$

output

index over input connections

weight on ith input



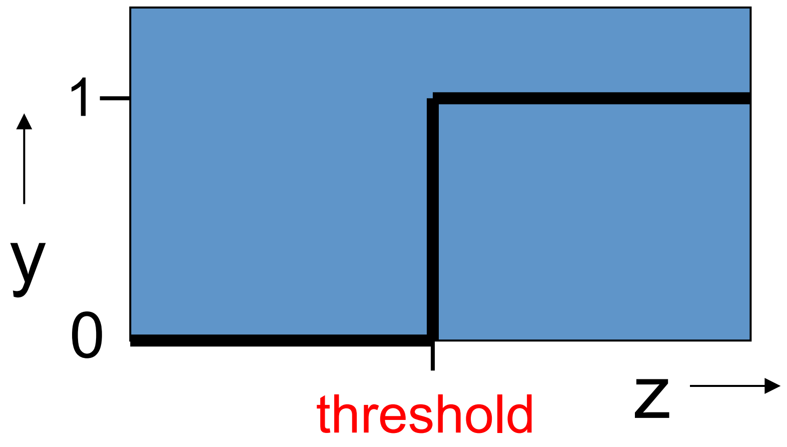
Neural Networks - Binary threshold neurons

McCulloch-Pitts (1943): influenced Von Neumann!

- First compute a weighted sum of the inputs from other neurons
- Then send out a fixed size spike of activity if the weighted sum exceeds a threshold.
- McCulloch & Pitts: each spike is like the truth value of a proposition and each neuron combines truth values to compute the truth value of another proposition.

$$z = \sum_i x_i w_i$$

$$y = \begin{cases} 1 \text{ if } & z \geq \theta \\ 0 \text{ otherwise} \end{cases}$$

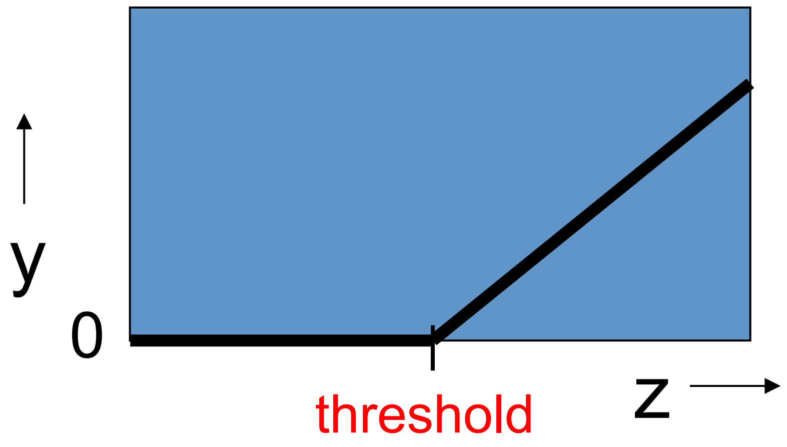


Neural Networks – Linear threshold neurons

These have a confusing name.
They compute a linear weighted sum of their inputs
The output is a non-linear function of the total input

$$z_j = b_j + \sum_i x_i w_{ij}$$

$$y_j = \begin{cases} z_j \text{ if } z_j \geq 0 \\ 0 \text{ otherwise} \end{cases}$$



Neural Networks – Sigmoid neurons

These give a real-valued output that is a smooth and bounded function of their total input.

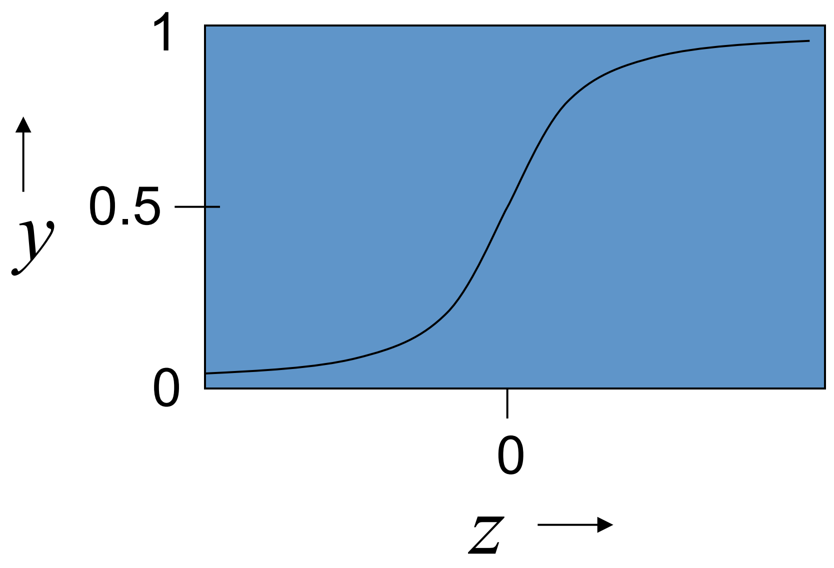
- Typically they use the logistic function
- They have nice derivatives which make learning easy.

•If we treat y as a probability of producing a spike, we get stochastic binary neurons.

The image cannot be displayed. Your computer may not have enough memory to open the image, or the image may have been corrupted. Restart your computer, and then open the file again. If the red x still appears, you may have to delete the image and then insert it again.

$$z = b + \sum_i x_i w_i$$

$$y = \frac{1}{1 + e^{-z}}$$



Neural Networks - A very simple way to recognize handwritten shapes

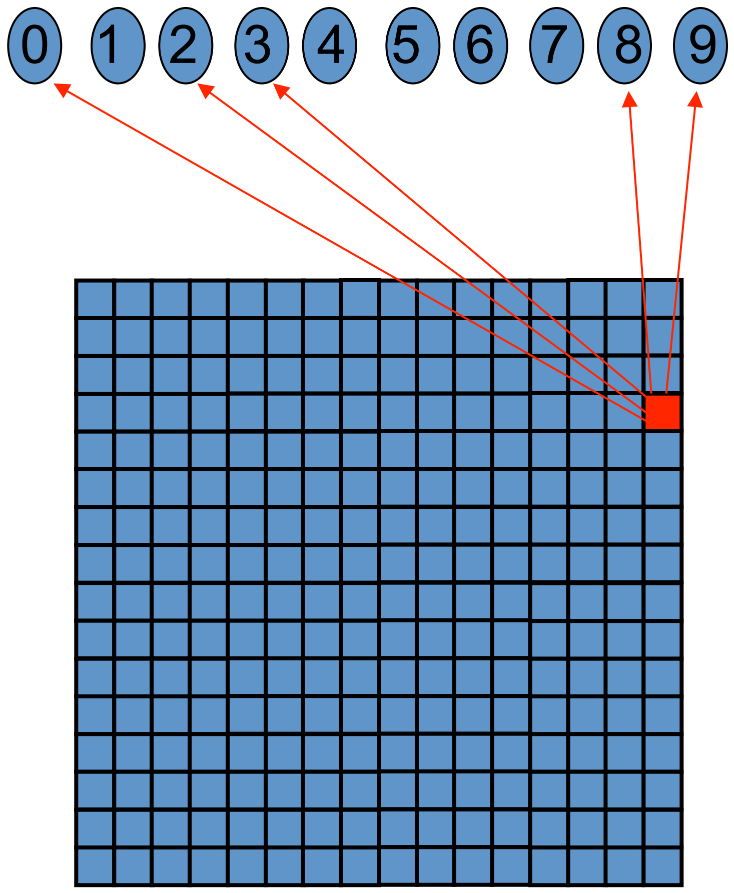
Consider a neural network with two layers of neurons.

- neurons in the top layer represent known shapes.
- neurons in the bottom layer represent pixel intensities.

A pixel gets to vote if it has ink on it.

- Each inked pixel can vote for several different shapes.

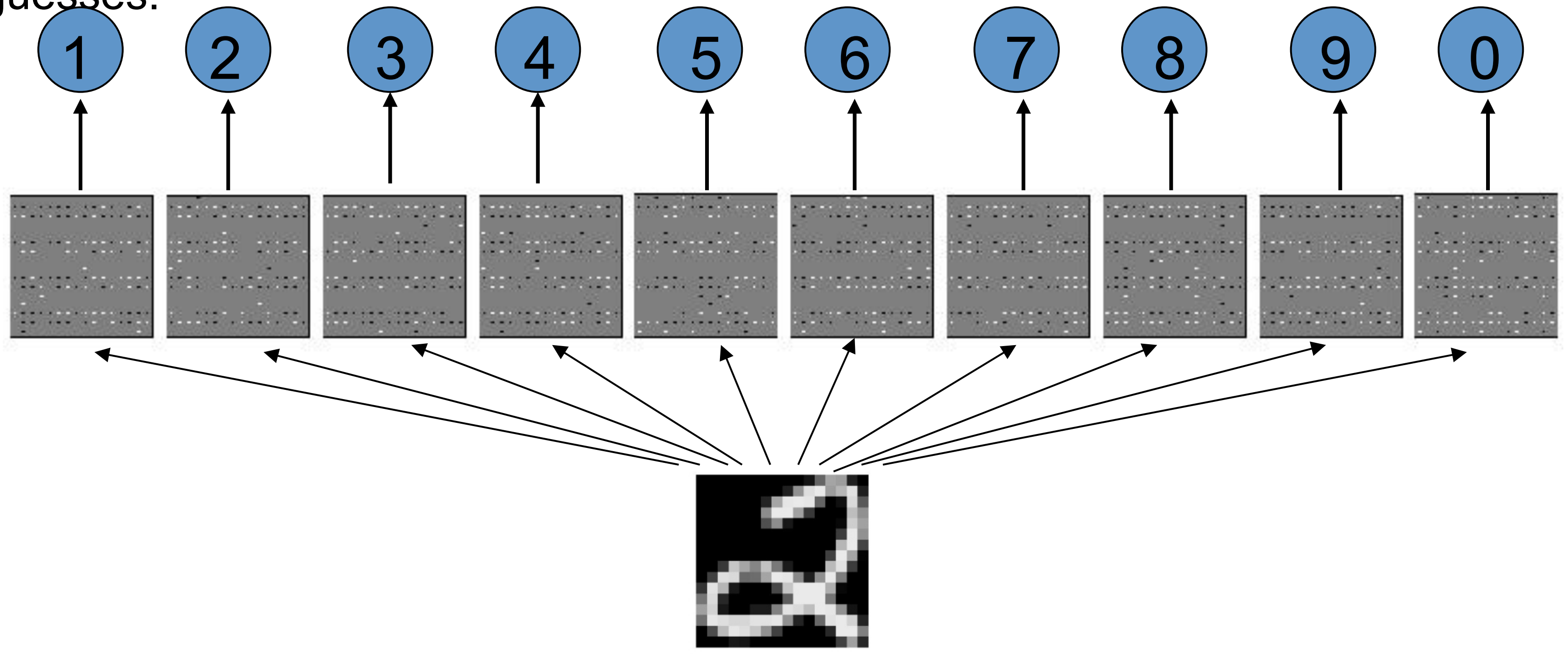
The shape that gets the most votes wins.



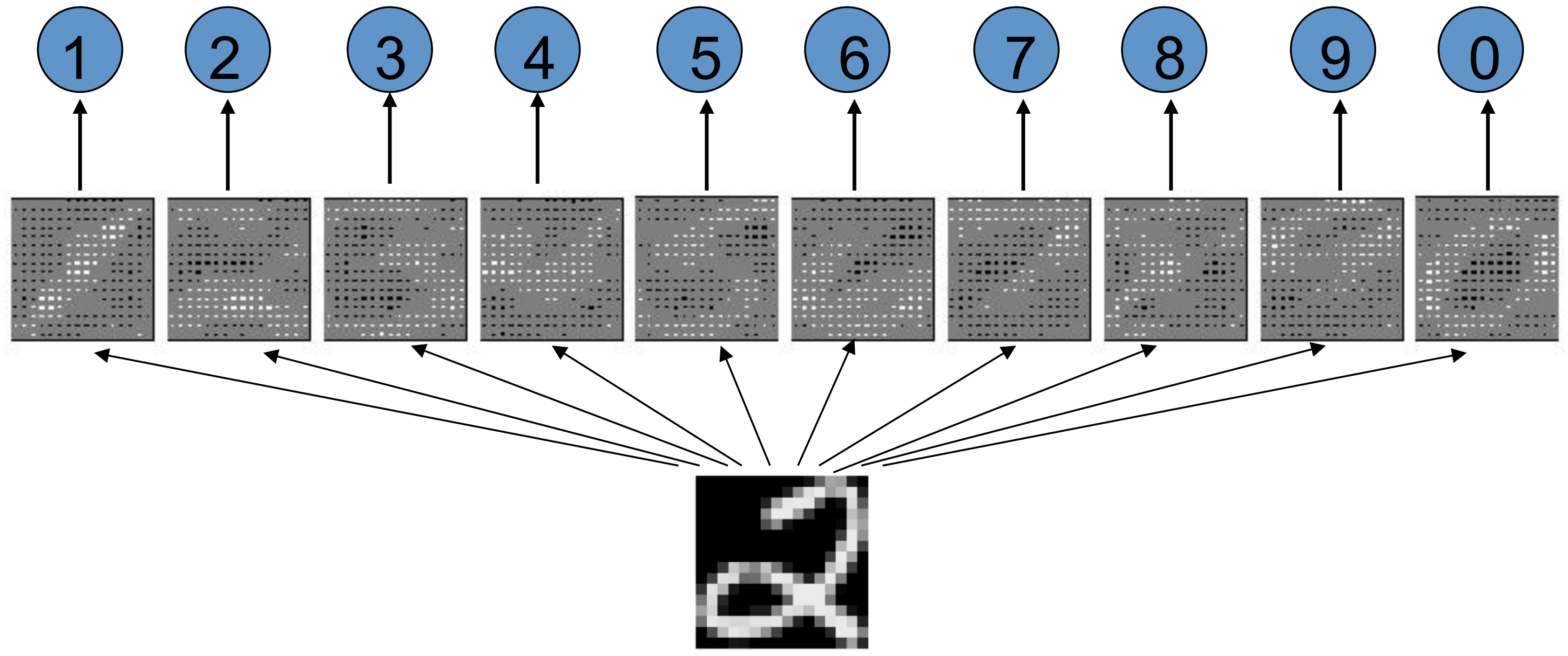
Neural Networks – How to learn weights

Show the network an image and increment the weights from active pixels to the correct class.

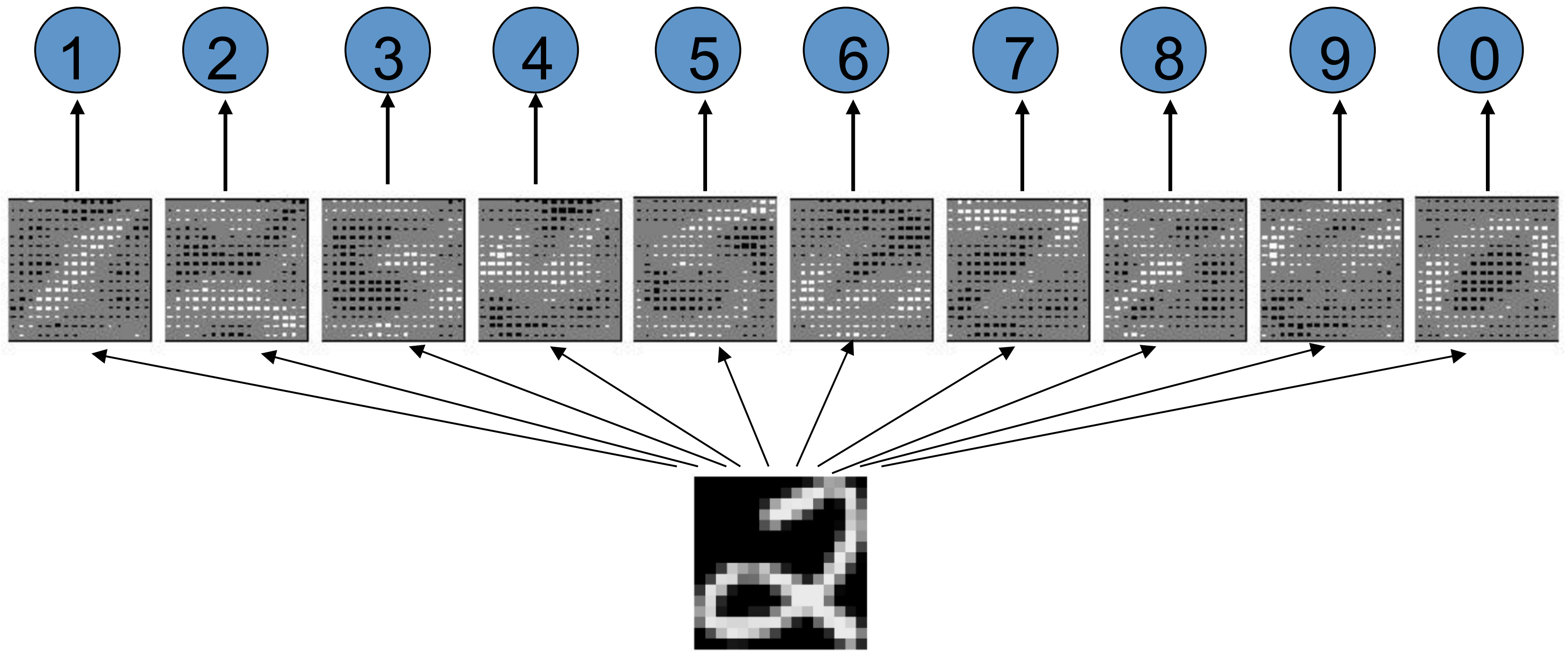
Then decrement the weights from active pixels to whatever class the network guesses.



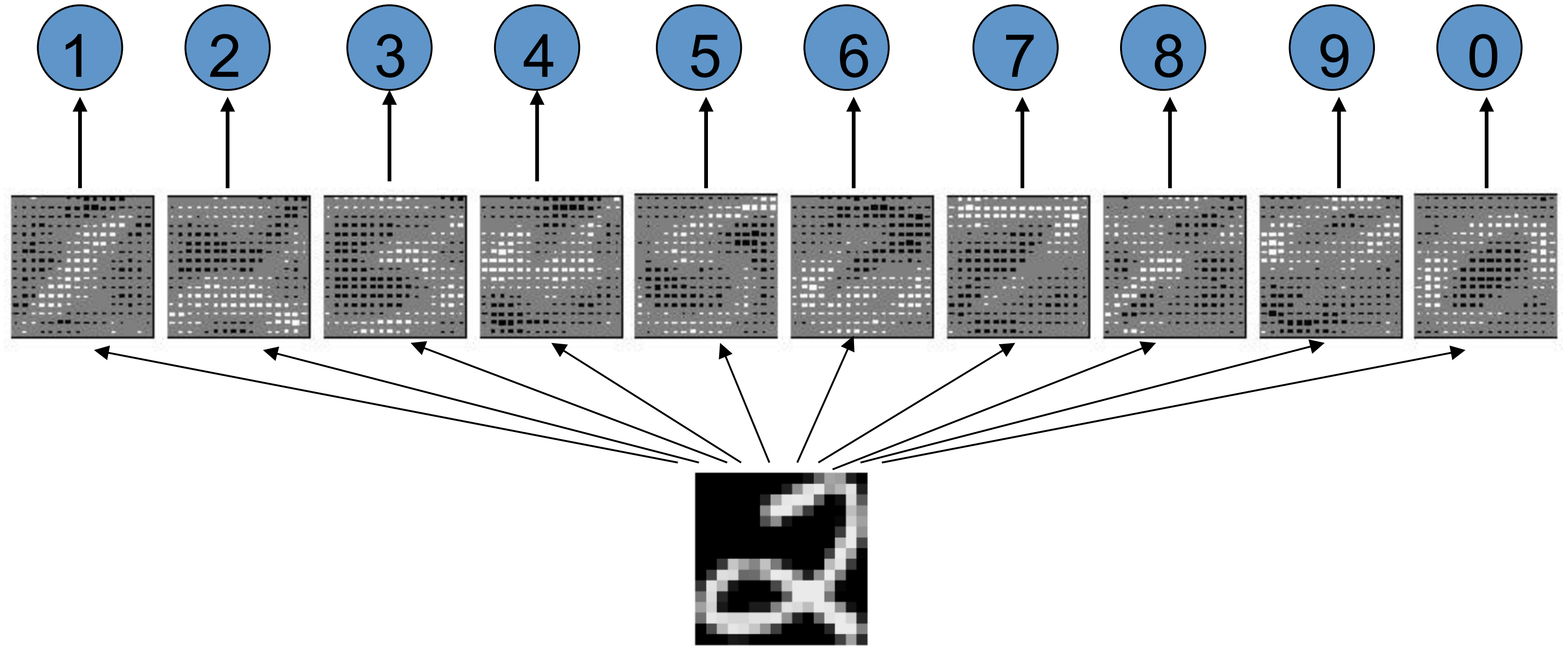
The image



The image

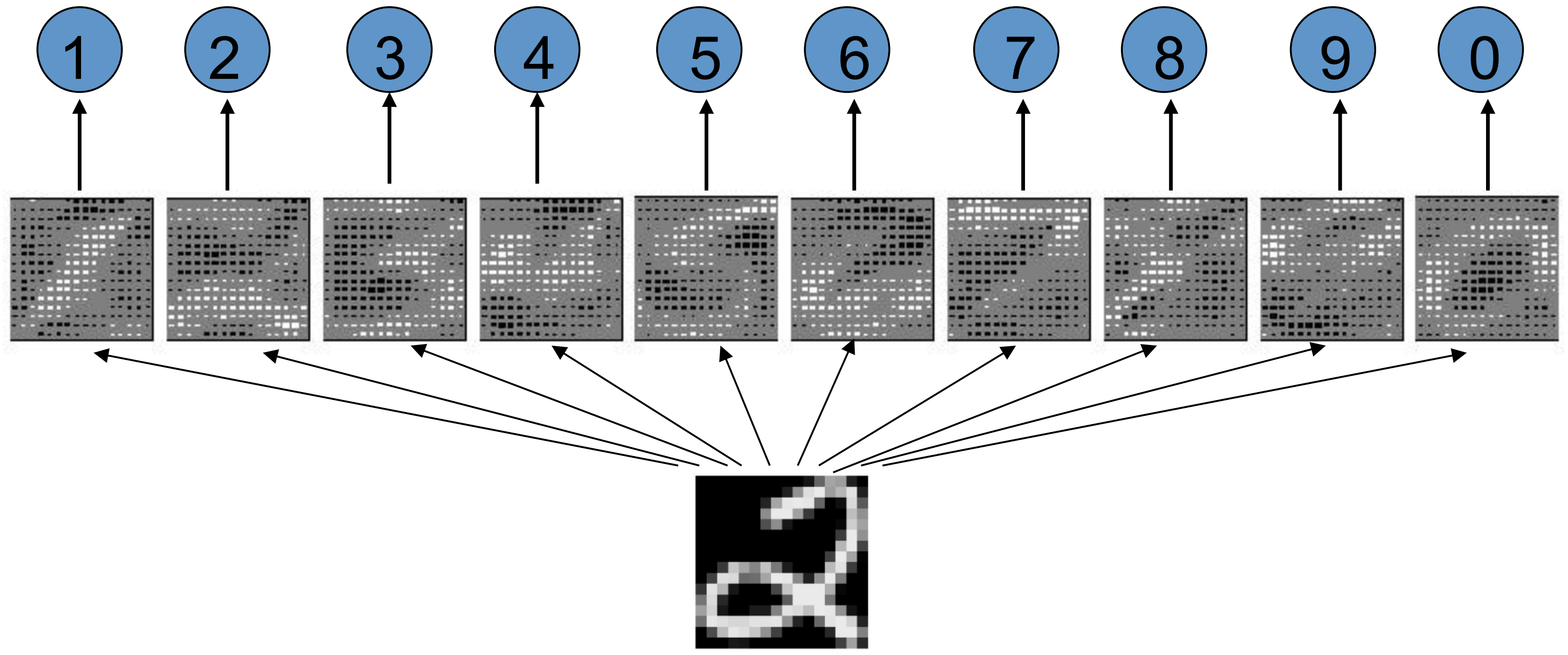


The image


1
2
3
4
5
6
7
8
9
0

Neural Networks – The learned weights

The precise details of the learning algorithm will be explained in future.



The image

Neural Networks – Why the simple system does not work

A two layer network with a single winner in the top layer is equivalent to having a rigid template for each shape.

- The winner is the template that has the biggest overlap with the ink.

The ways in which shapes vary are much too complicated to be captured by simple template matches of whole shapes.

- To capture all the allowable variations of a shape we need to learn the features that it is composed of.

Neural Networks – Types of Connectivity

Feedforward networks

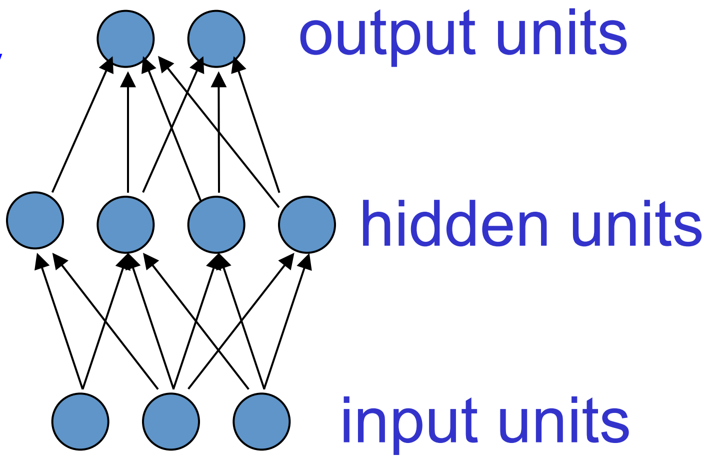
–These compute a series of transformations

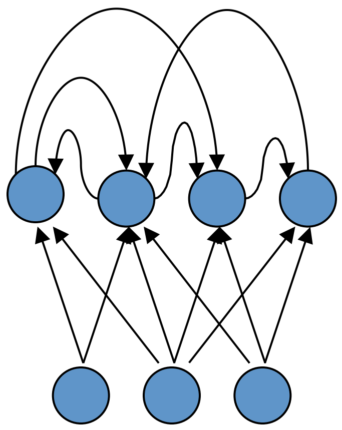
–Typically, the first layer is the input and the last layer is the output.

Recurrent networks

–These have directed cycles in their connection graph. They can have complicated dynamics.

–More biologically realistic.





Neural Networks – Types of Learning Task

Supervised learning

- Learn to predict output when given input vector
    - Who provides the correct answer?

Reinforcement learning

- Learn action to maximize payoff
    - Not much information in a payoff signal
    - Payoff is often delayed

Unsupervised learning

- Create an internal representation of the input e.g. form clusters; extract features
    - How do we know if a representation is good?

Concluding Remarks

THE PARADOX OF LIFE

A bit beyond perception's reach
I sometimes believe I see
that Life is two locked boxes, each
containing the other's key.