UNIVERSITY AT ALBANY
STATE UNIVERSITY OF NEW YORK

CSI 436/536 (Fall 2024)

Machine Learning

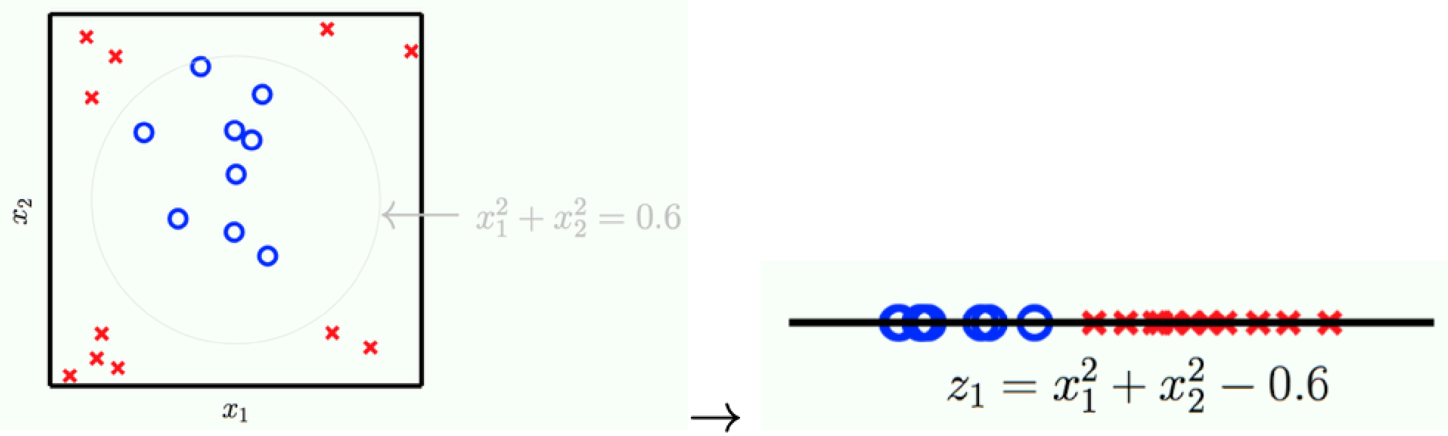
Lecture 17: Neural Network and Deep Learning

Chong Liu

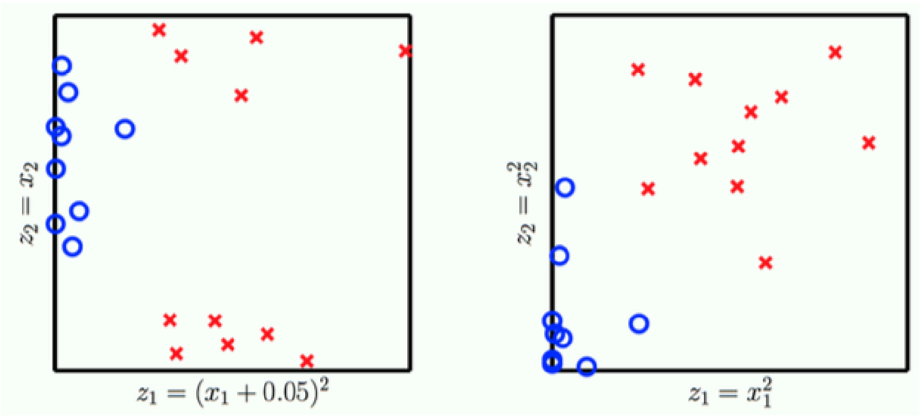
Assistant Professor of Computer Science

Nov 12, 2024

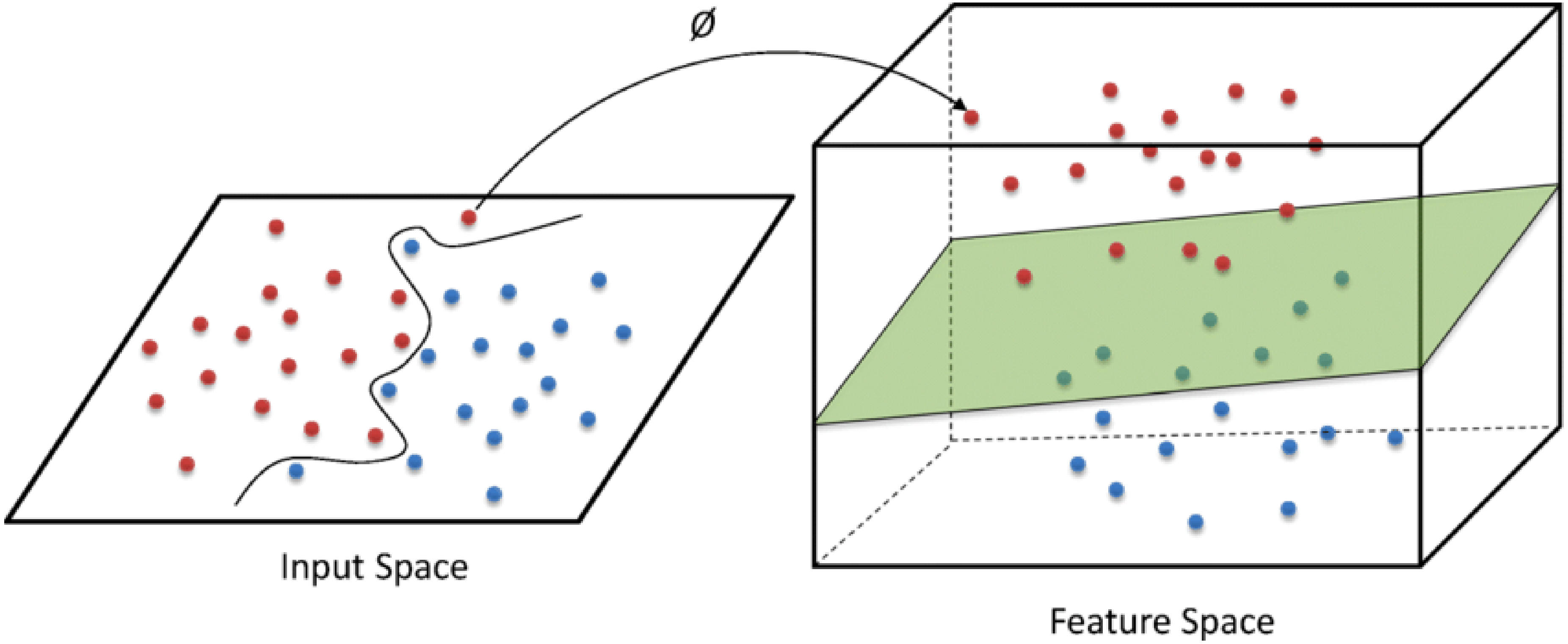
Recap: Many ways to transform features



And many other would work ...



Recap: It is easier to linearly classify the data in higher dimensions



Today

- Neural network

- Deep learning

Example of neural network: AlexNet (2012) – starting point of deep learning

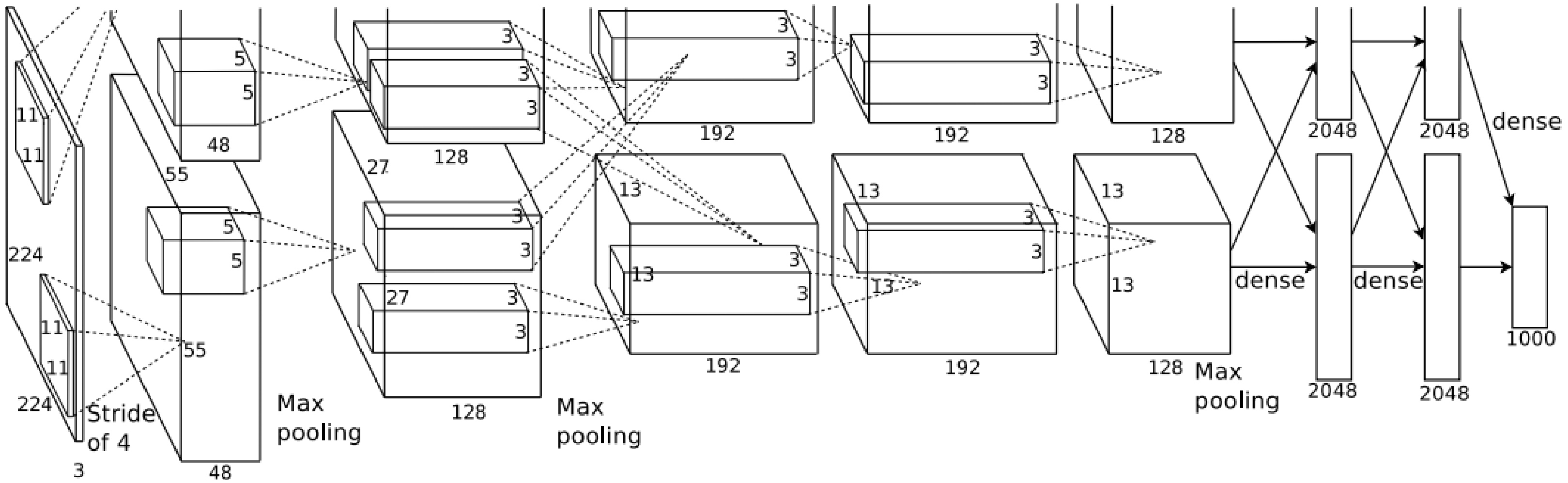
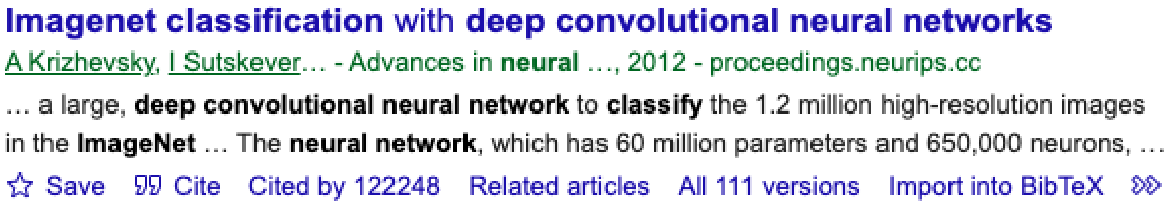


Figure 2: An illustration of the architecture of our CNN, explicitly showing the delineation of responsibilities between the two GPUs. One GPU runs the layer-parts at the top of the figure while the other runs the layer-parts at the bottom. The GPUs communicate only at certain layers. The network's input is 150,528-dimensional, and the number of neurons in the network's remaining layers is given by 253,440–186,624–64,896–64,896–43,264–4096–4096–1000.



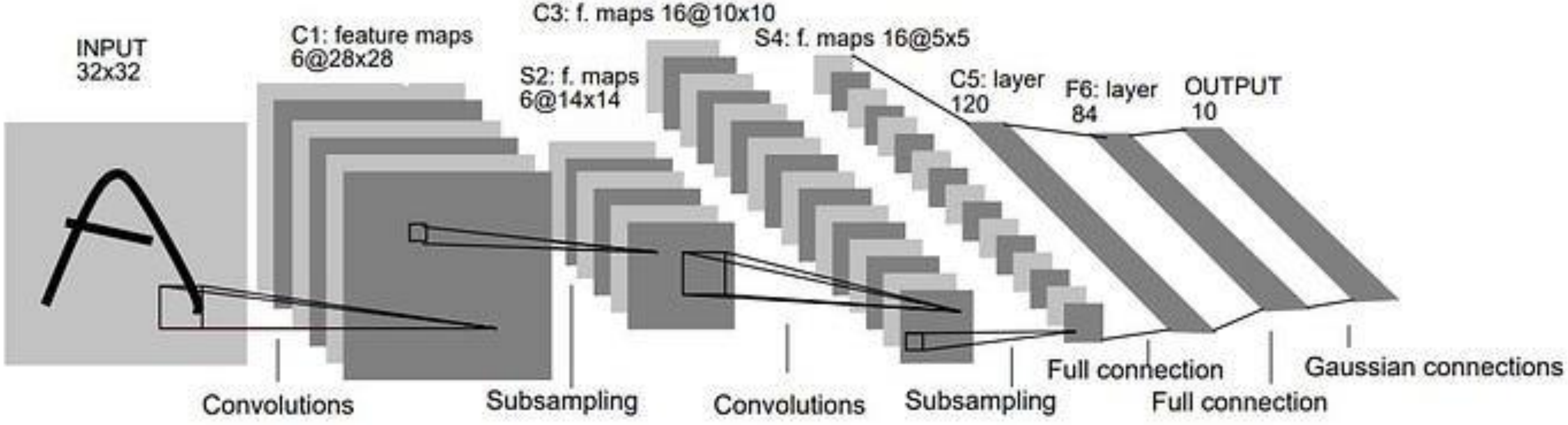
Imagenet classification with **deep convolutional neural networks**

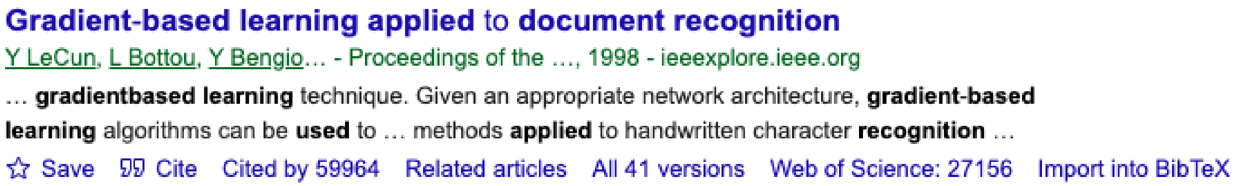
A Krizhevsky, I Sutskever… - Advances in **neural** …, 2012 - proceedings.neurips.cc

… a large, **deep convolutional neural network** to **classify** the 1.2 million high-resolution images in the **ImageNet** … The **neural network**, which has 60 million parameters and 650,000 neurons, …

☆ Save 99 Cite Cited by 122248 Related articles All 111 versions Import into BibTeX »

LeNet (1998)





Gradient-based learning applied to **document recognition**

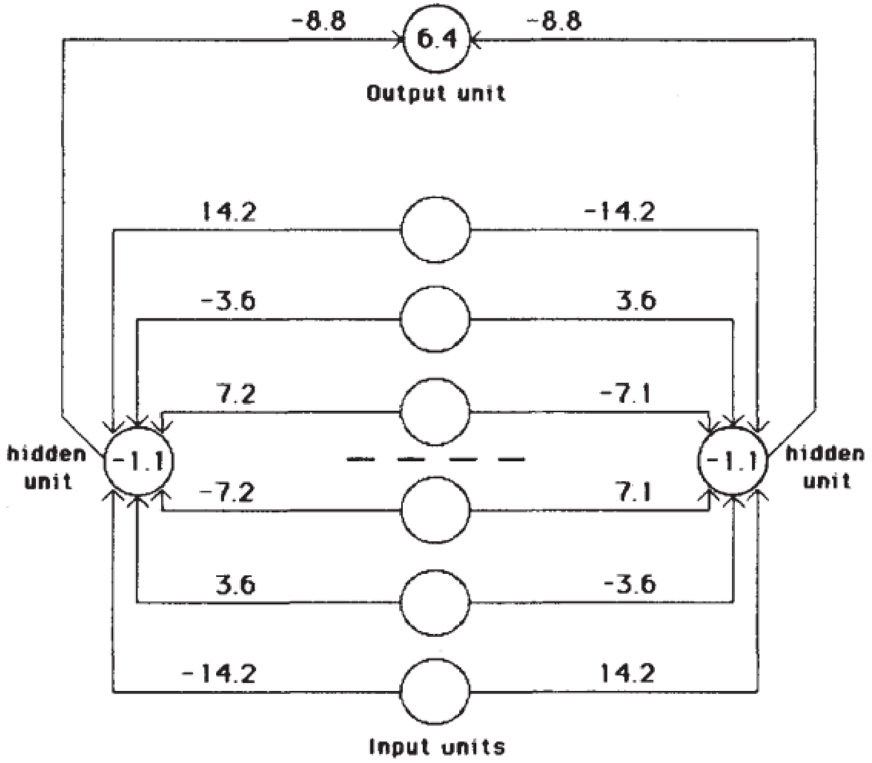
Y LeCun, L Bottou, Y Bengio… - Proceedings of the …, 1998 - ieeexplore.ieee.org

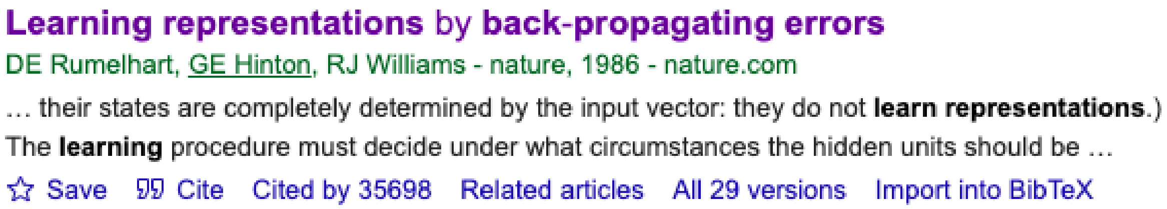
… **gradientbased learning** technique. Given an appropriate network architecture, **gradient-based learning** algorithms can be **used** to … methods **applied** to handwritten character **recognition** …

☆ Save 99 Cite Cited by 59964 Related articles All 41 versions Web of Science: 27156 Import into BibTeX

Rumelhart, Hinton, Williams (1986)

- One layer of a feedforward neural networks





Learning representations by **back-propagating errors**

DE Rumelhart, GE Hinton, RJ Williams - nature, 1986 - nature.com

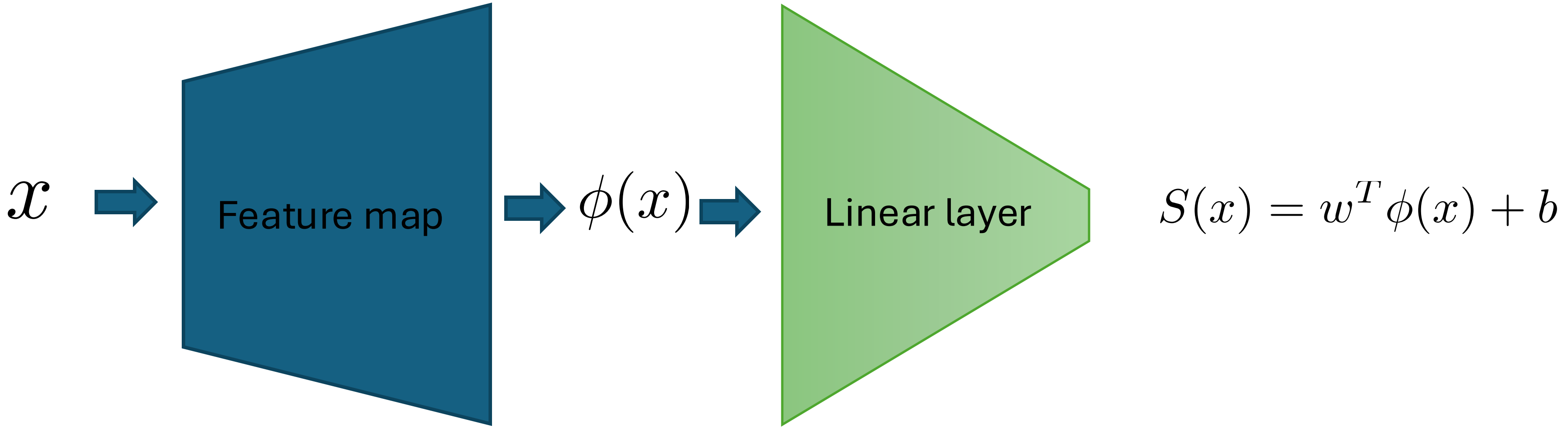
… their states are completely determined by the input vector: they do not **learn representations**.) The **learning** procedure must decide under what circumstances the hidden units should be …

☆ Save 99 Cite Cited by 35698 Related articles All 29 versions Import into BibTeX

It goes back even further…

- 1943 Pitts and McCulloch: Perceptron model to mimic the brain
- 1956: Rosenblatt's Perceptron Implementation
- 1960s:
    - Ivakhnenko and Lapa: Multi-layer Perceptron (going deeper)
    - Dreyfus: Backpropagation for training (not yet the same as SGD)
    - Amari: Use SGD for training MLPs (separating non-linearly separable patterns)
- 1970s:
    - Fukushima: Convolutional Neural Networks for images
- 1982:
    - Werbos: Modern day backpropagation / SGD

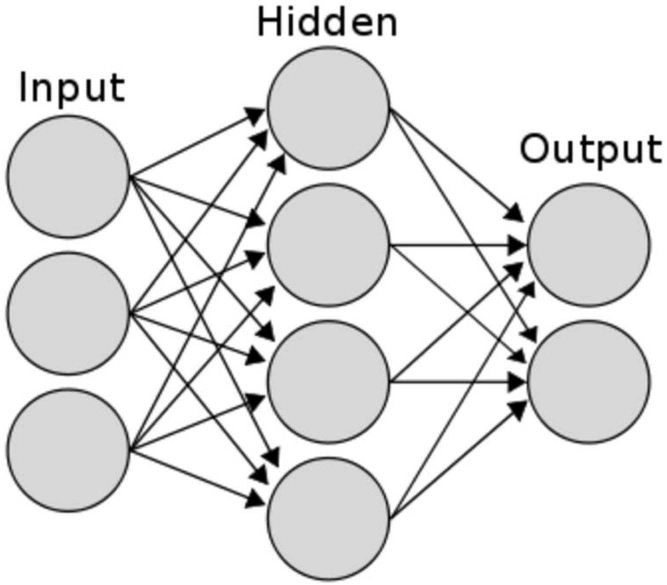
From kernels to neural networks



Two-layer neural networks

- Neural network: $S(x) \ = \ w_2^T(W_1 x + \boldsymbol{b}_1) + b_2$
    - In-class exercise: is this linear of x?
    - Still a linear model at the end of the day, so let’s add a nonlinearity σ!

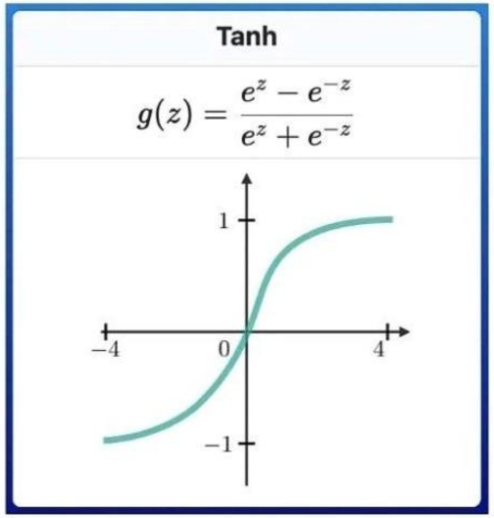
- Two-layer MLP: $S(x) \ = \ w_2^T\sigma(W_1 x + \boldsymbol{b}_1) + b_2$
    - Suppose σ is a non-linear function
    - In-class exercise: is this linear of x?
    - Linear model w.r.t. to a learnable feature map



- RBF-kernel: $S(x) = w_2^T \exp\bigl(-\gamma(W_1 x + \boldsymbol{b}_1)\bigr) + b_2$

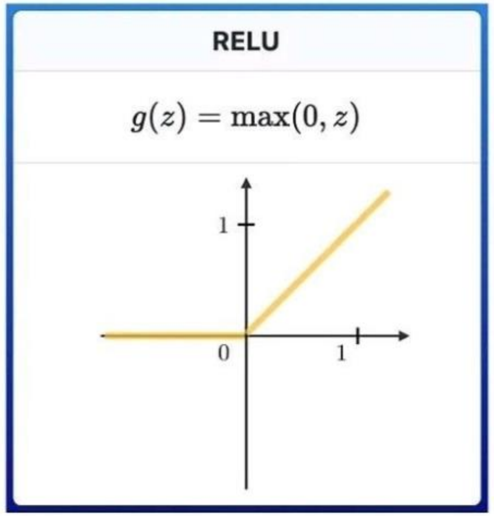
Choices of activation function

- Activation function must be non-linear!



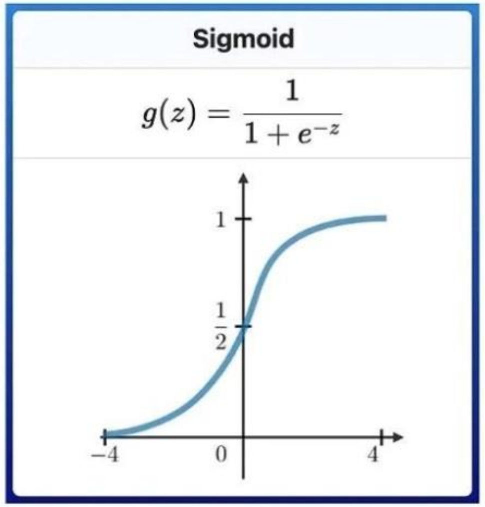
Tanh

$$g(z) = \frac{e^z - e^{-z}}{e^z + e^{-z}}$$



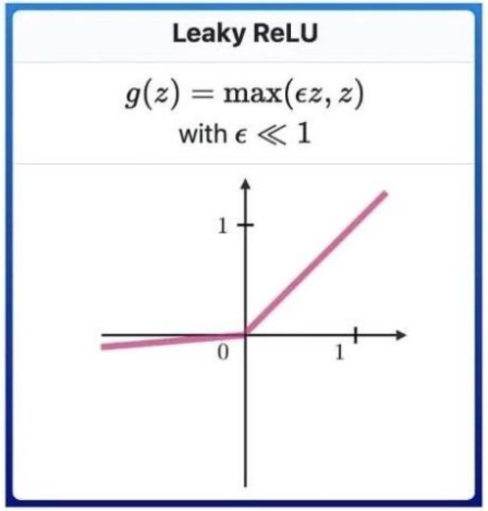
RELU

$$g(z) = \max(0, z)$$



Sigmoid

$$g(z) = \frac{1}{1 + e^{-z}}$$

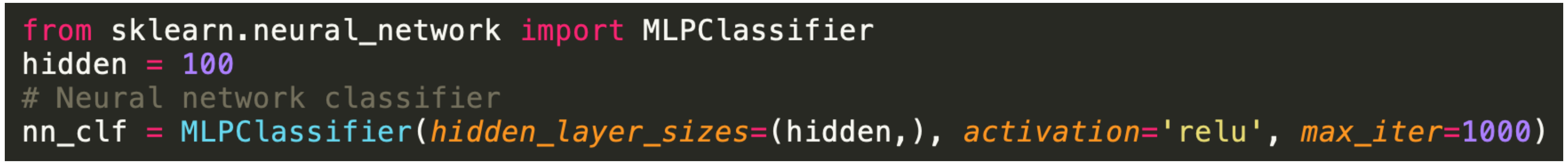


Leaky ReLU

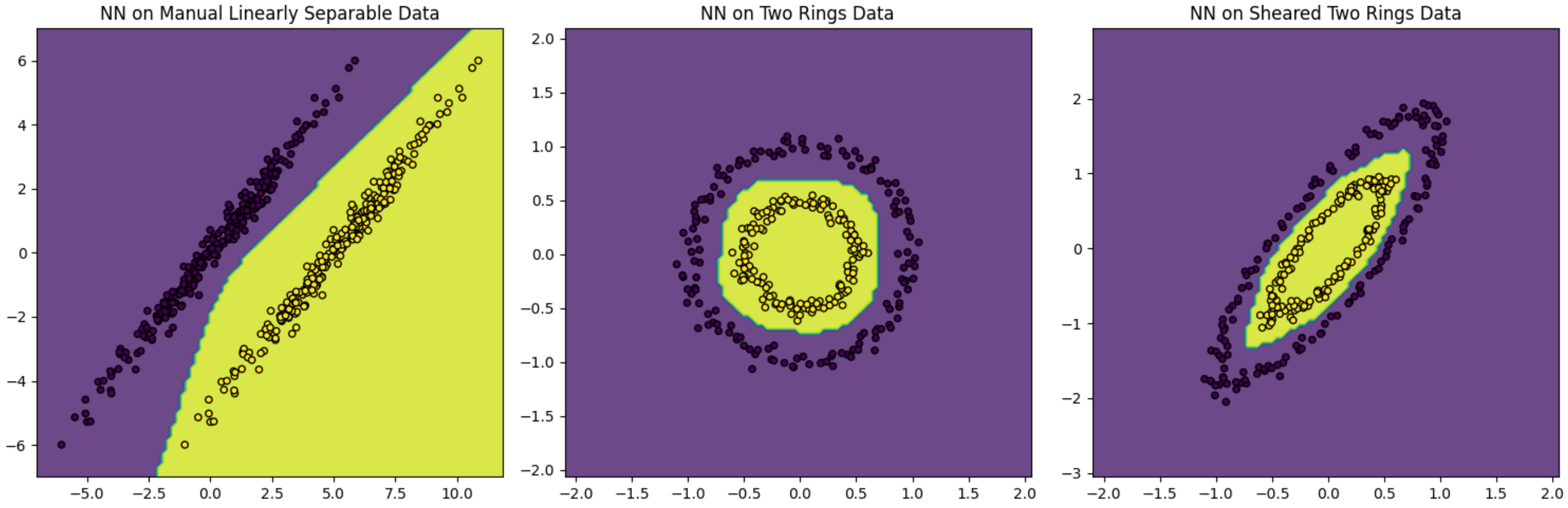
$$g(z) = \max(\epsilon z, z)$$

with $\epsilon \ll 1$

Results of fitting MLPs on our three examples



```python
from sklearn.neural_network import MLPClassifier
hidden = 100
# Neural network classifier
nn_clf = MLPClassifier(hidden_layer_sizes=(hidden,), activation='relu', max_iter=1000)
```



NN on Manual Linearly Separable Data

NN on Two Rings Data

NN on Sheared Two Rings Data

Deep Learning

- Deep learning model is usually referred to deep neural network
    - Many many layers

- Some useful facts about deep learning to know:
    1. Non-linear activation function.
    2. Feature expansion technique but with learned features.
    3. You choose a deep learning model for constructing hypothesis classes that are suitable for your problem.
    4. Training process requires a lot of computational resources.

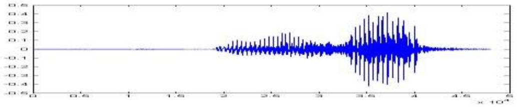
You can use deep learning for all kinds of ML problems: classification, regression, clustering, dimension reduction etc..

- Deep learning provides a learnable function approximation

- Different kinds of architecture (like LEGO blocks) are designed to address different challenges in different kind of problems:
  - Feedforward neural network
  - Recurrent neural network
  - Boltzmann machine
  - Convolutional neural network
  - Graph Neural Networks
  - Transformers
  - …

Learning ≈ Configuring the learnable function so it behaves as instructed.

- Speech Recognition

$f($  $)=$ “Hello!”

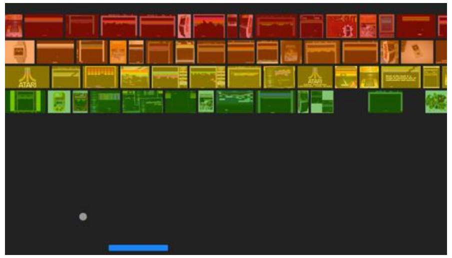
- Handwritten Recognition

$f($ $)=$ “2”

- Weather forecast

$f($ ⛅ Thursday $)=$ “ 🌧 Saturday”

- Play video games

$f($  $)=$ “move left”

Generally speaking, you need to make decisions about

- Which loss function to use
  - For regression, classification, clustering, dimension reduction, but also ranking, recommendation, and others…
- What type of neural network to use
  - Images
  - Text
  - Graphs (node and edges)
  - Time series
  - Decide on the hyperparameters: Depth, Width, Number of hidden units…
- How to train the neural network?
  - Initialization of weights: iid random? Rescale or not?
  - Optimizer to use: SGD, ADAM, etc…
- How to collect, pre-process the data…

Modern neural networks are very complicated – ResNet (2016)

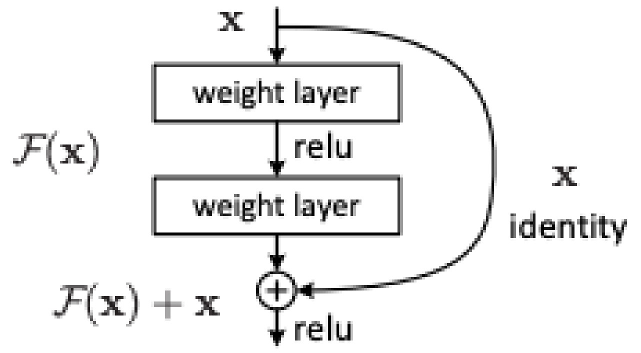
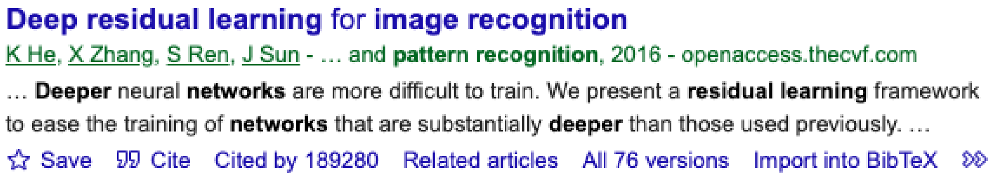

x
weight layer
$\mathcal{F}(\mathbf{x})$
relu
weight layer
x
identity
$\mathcal{F}(\mathbf{x}) + \mathbf{x}$
relu

Figure 2. Residual learning: a building block.

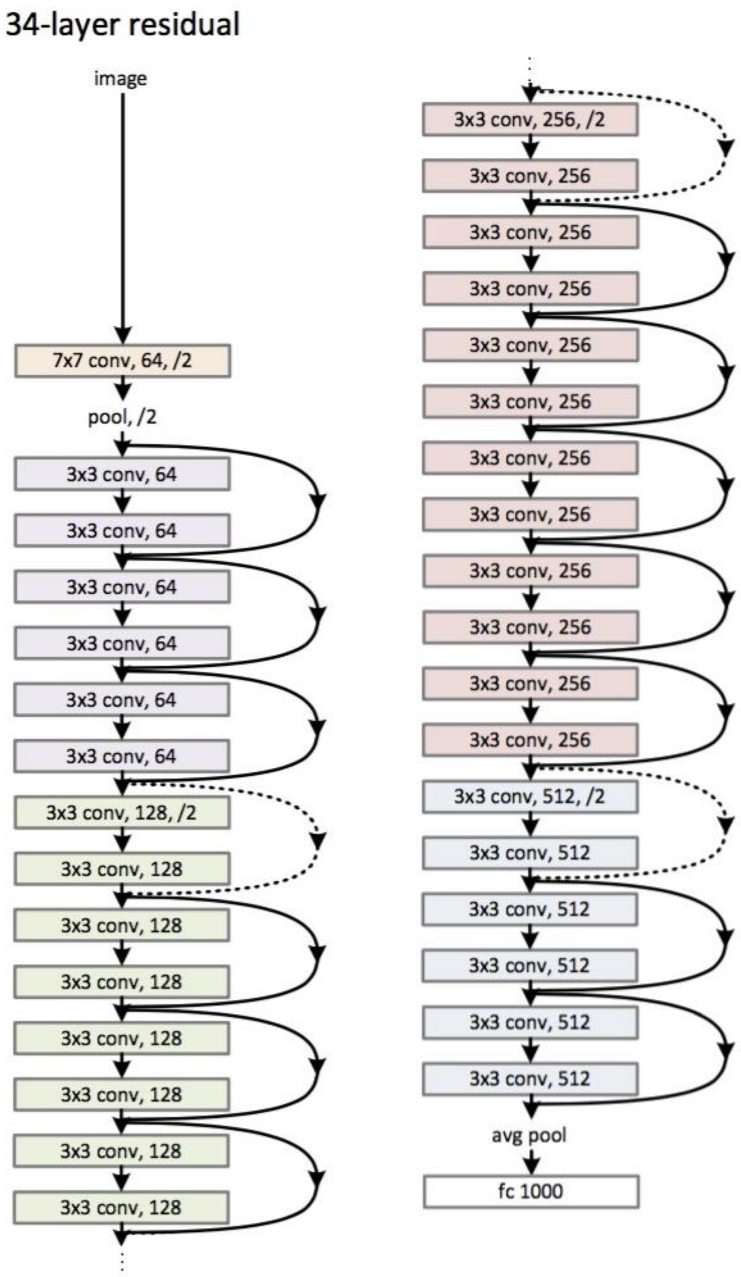


Deep residual learning for **image recognition**

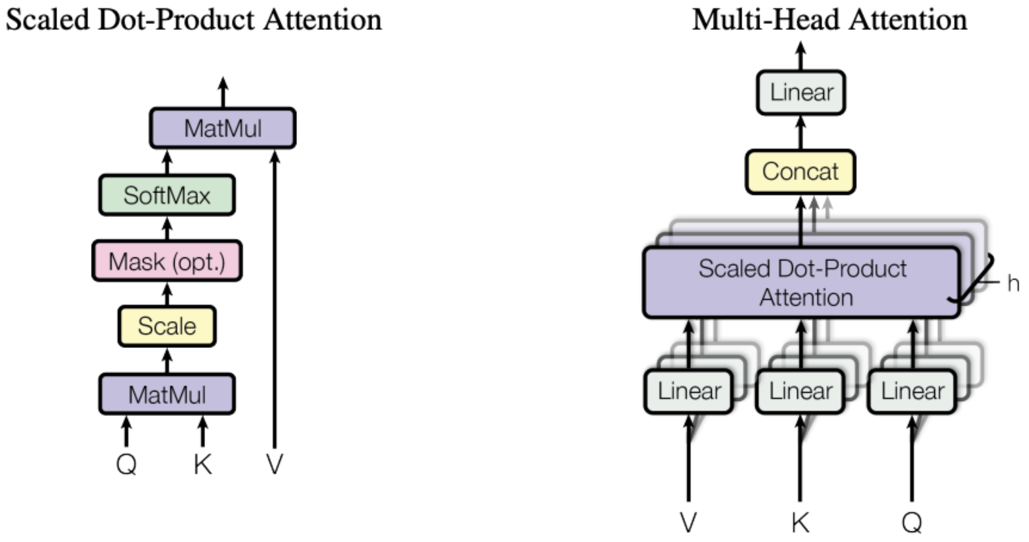
K He, X Zhang, S Ren, J Sun - … and **pattern recognition**, 2016 - openaccess.thecvf.com

… **Deeper** neural **networks** are more difficult to train. We present a **residual learning** framework to ease the training of **networks** that are substantially **deeper** than those used previously. …

☆ Save 99 Cite Cited by 189280 Related articles All 76 versions Import into BibTeX »



Modern neural networks are very complicated – Transformer (2017)



Scaled Dot-Product Attention

Multi-Head Attention

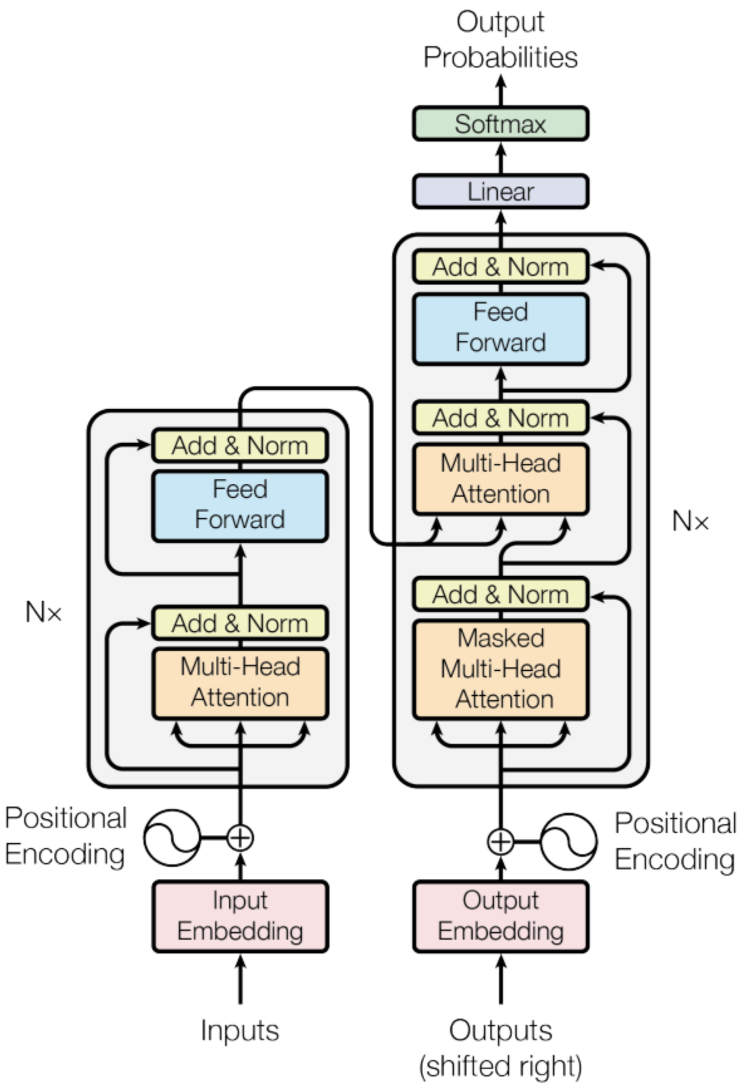
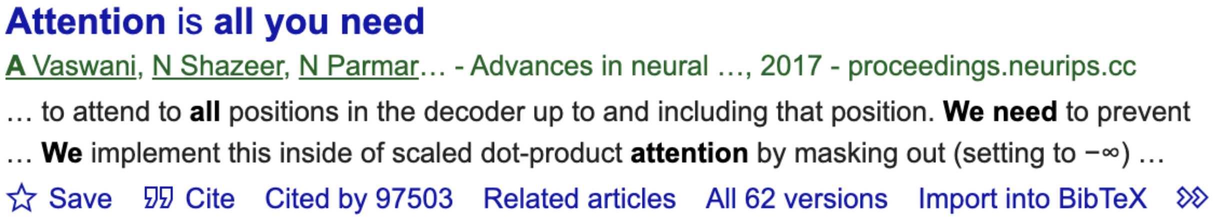


Figure 1: The Transformer - model architecture.



Attention is all you need

A Vaswani, N Shazeer, N Parmar… - Advances in neural …, 2017 - proceedings.neurips.cc

… to attend to **all** positions in the decoder up to and including that position. **We need** to prevent … **We** implement this inside of scaled dot-product **attention** by masking out (setting to −∞) …

Solution to this? Brute-force computation with autograd and GPUs

- **Autograd**: basically chain rules, can be automated.
  - Design networks such that every block is differentiable.

- **Faster Computation**:
  - Well-packaged Deep Learning Farmework: Write Python wrapper code but running C++ underneath
  - Parallel computing: Numerical linear algebra and GPUs, scientific computing, supercomputing centers.
  - Distributed computing: Cloud computing, Map-Reduce, federated learning

- Popular tools (there are many more of these):

PyTorch

tensorflow

JAX: Autograd and XLA

Summary

- Neural network
  - Learning with neural network == fitting a neural network function
  - How to build a strong neural network with great learning ability?
    - Non-linear activation function
    - More layers
- Deep learning
  - Deep learning models are deep neural networks with many many layers
  - Its training process requires a lot of resources
  - It can be used for all kinds of ML problems
- Create non-linear hypothesis:
  - Ensemble methods (bagging, boosting)
  - Neural network
- Transform feature representation:
  - Kernel methods
  - Neural network

Announcement

- Final exam
  - Monday December 16, 2024
  - 8:30-10am
  - Lecture center 5 (classroom)

- Homework
  - Homework 3 is due this Thursday!
  - Homework 4 will be released this Thursday