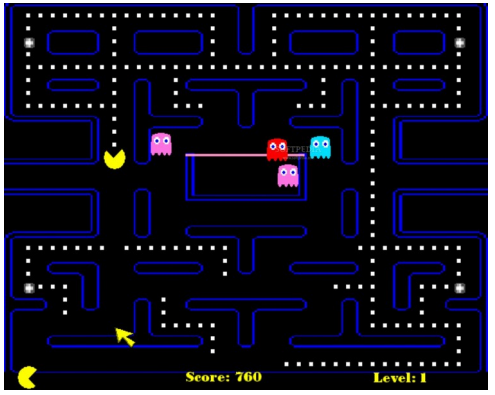
Brief Introduction to Neural Networks

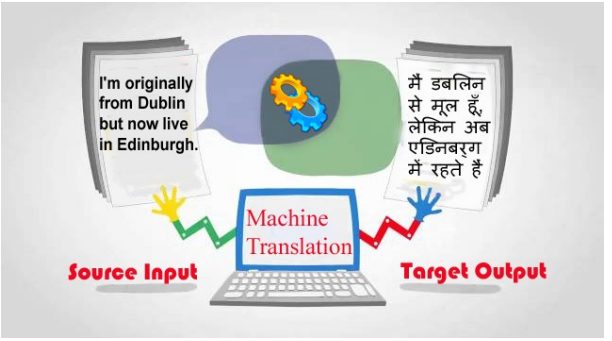
Content

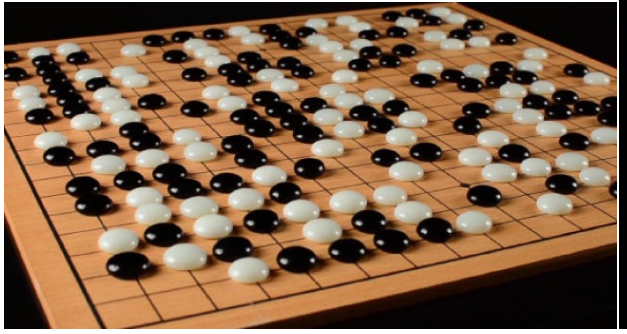
- What is a neural network?
- Typical architectures

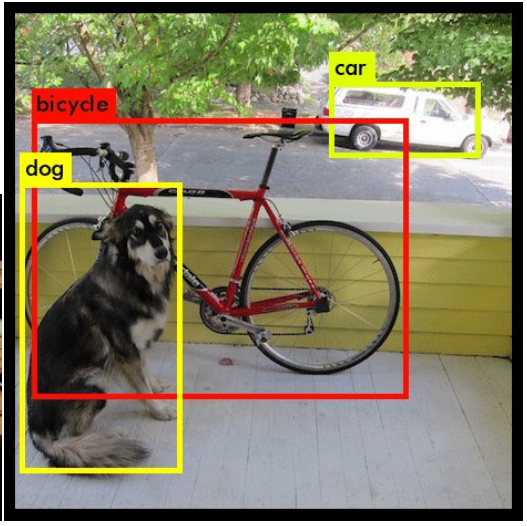
Neural Network =

- AI magic for
  - Image recognition
  - Machine translation
  - Playing Go
  - Playing Atari games
  - … anything?









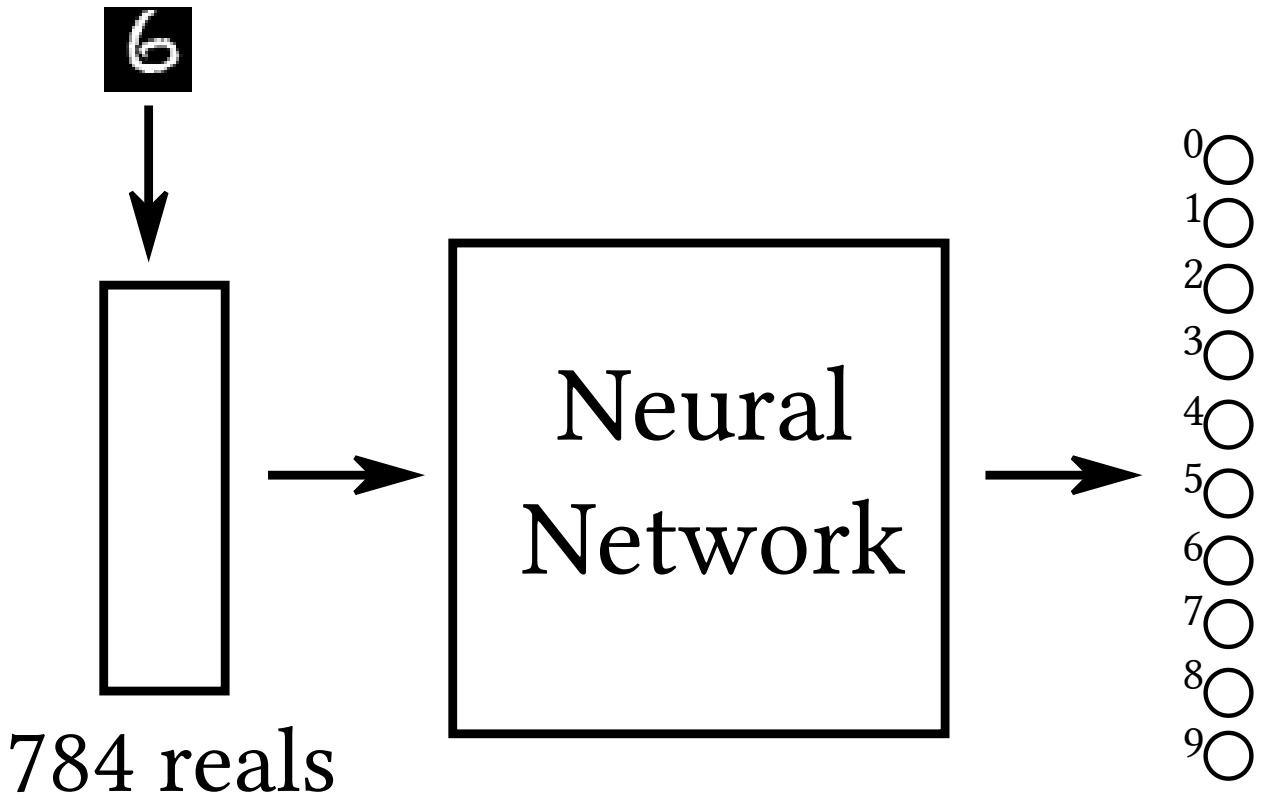
Neural Network =

A differentiable function
with many hidden parameters
trained by stochastic gradient descent

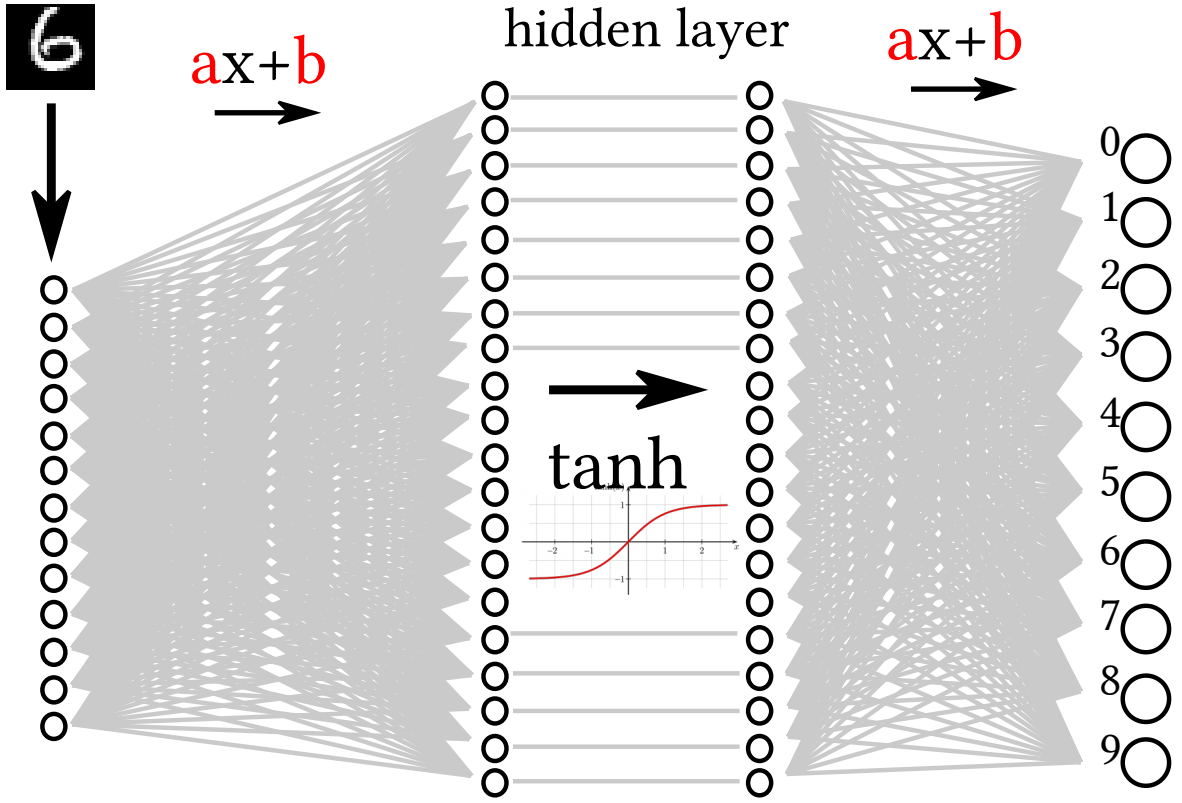
Example: MNIST

- MNIST is a dataset of labeled hand-drawn digits
- Every input is 28 x 28 grayscale image

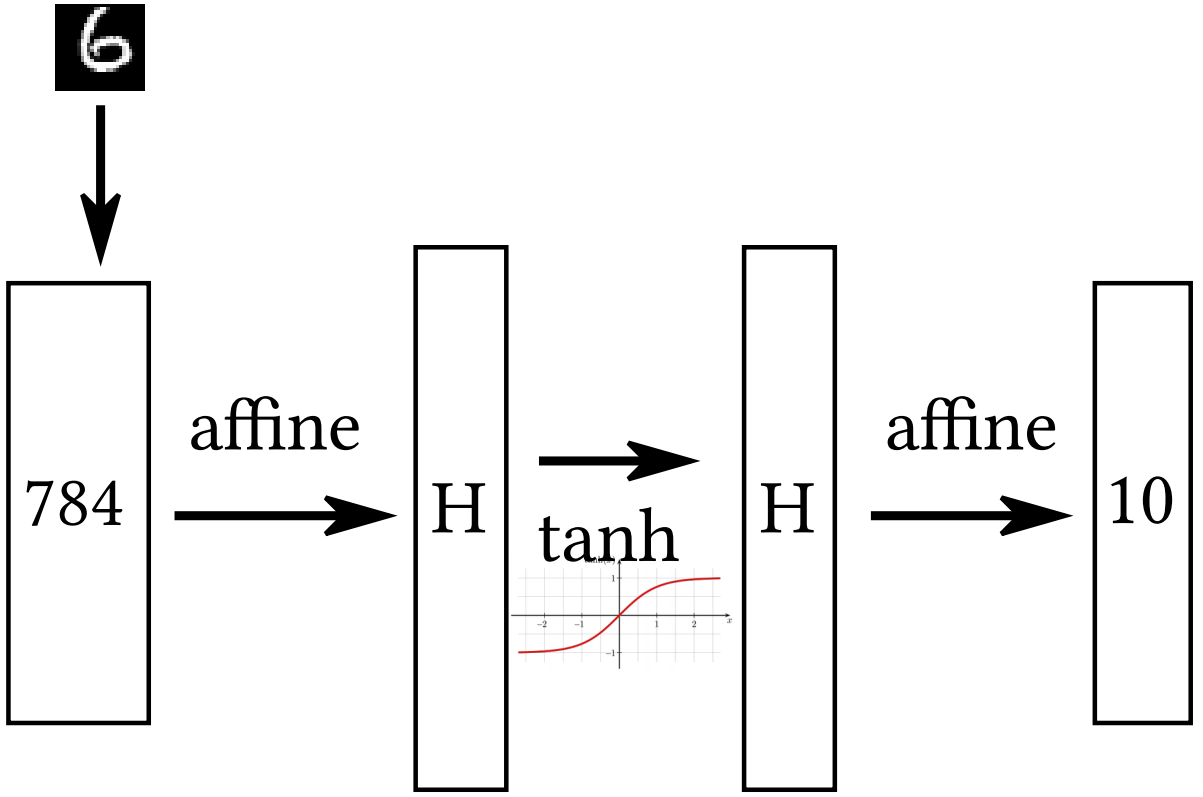
Example: MNIST



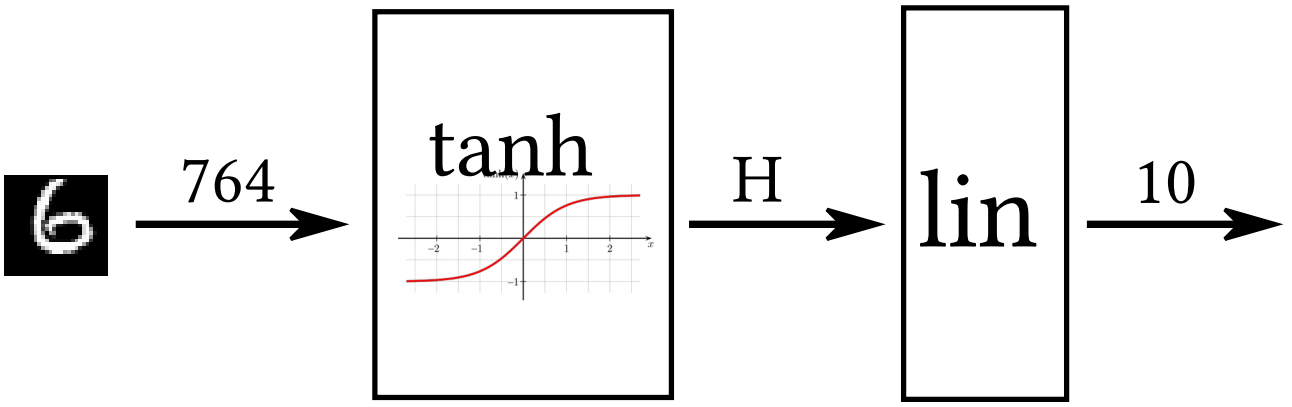
Example: MNIST



Example: MNIST



Example: MNIST



Training

- How to set the Network’s parameters

Training

- How to set the Network’s parameters
  - Training examples
  - Differentiable loss function
    - Input: the network output + the desired output
    - Output: “score” for the network (lower is better)
    - Usually “cross-entropy of softmax”

Training

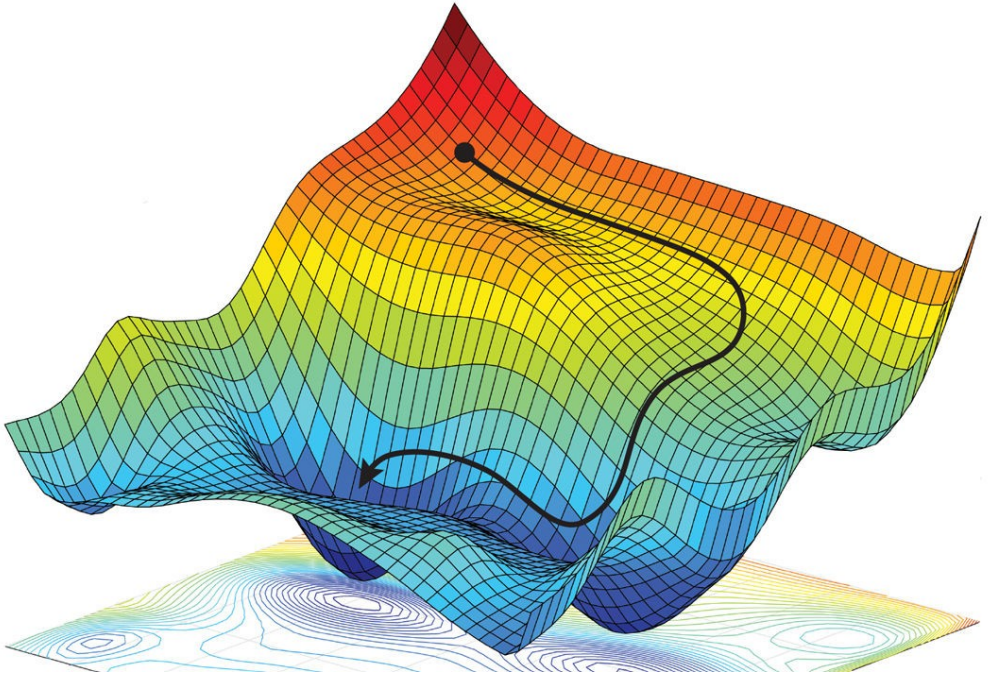
- How to set the Network's parameters
  - Training examples
  - Differentiable loss function
    - Input: the network output + the desired output
    - Output: "score" for the network (lower is better)
    - Usually "cross-entropy of softmax"
  - Minimize the average loss on the training examples

Training

- Minimize the average loss on the training examples

Training

- Minimize the average loss on the training examples
- By gradient descent



Training

- Minimize the average loss on the training examples
- By gradient descent
  - Efficient algorithm for gradient computation
    - “backpropagation”
  - Sampling training examples
    - “Stochastic” gradient descent

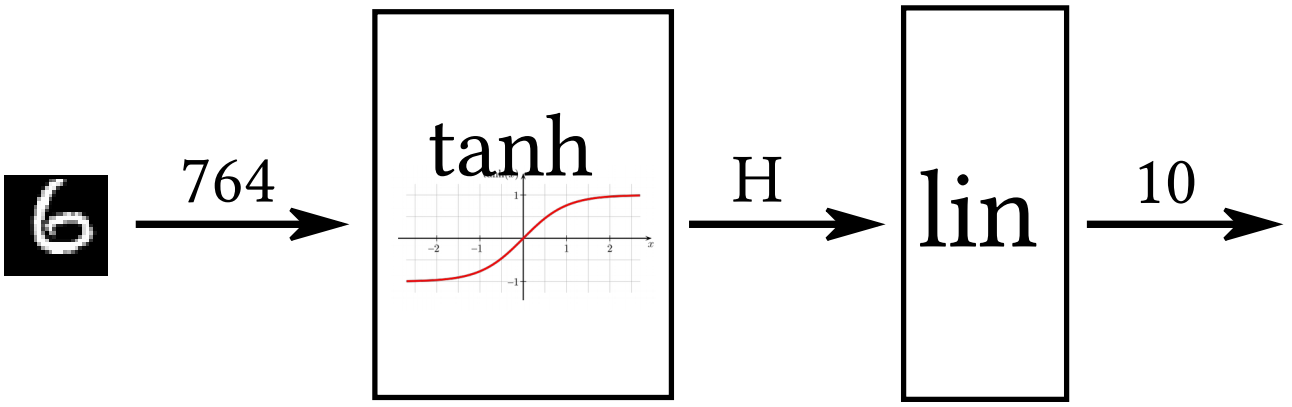
Neural Network =

A differentiable function
with many hidden parameters
trained by stochastic gradient descent

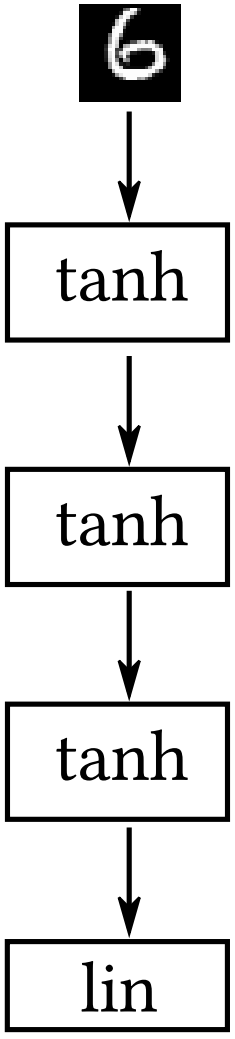
Archtectures

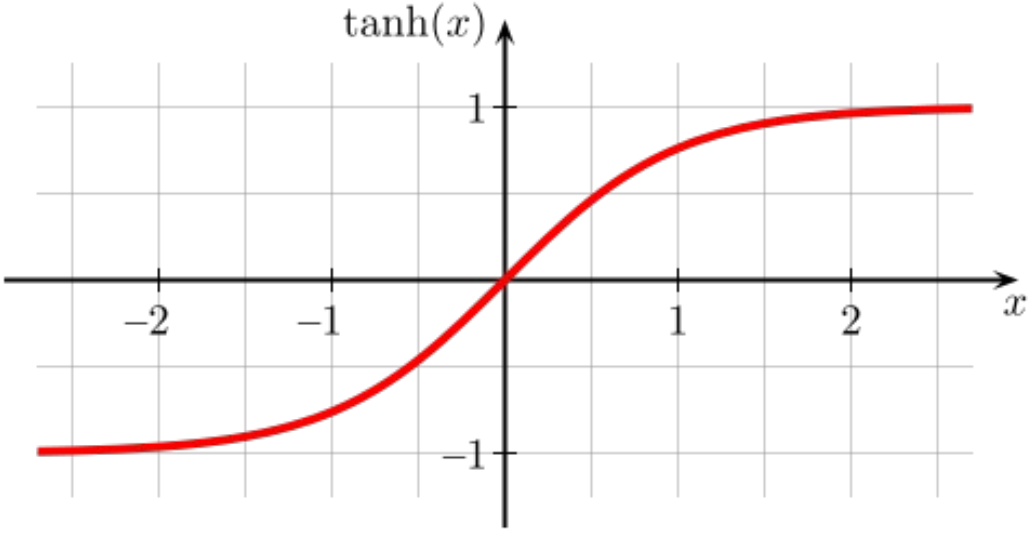
- More layers, ReLU activation
- Image processing – CNN
- Natural language processing – RNN

Example: MNIST



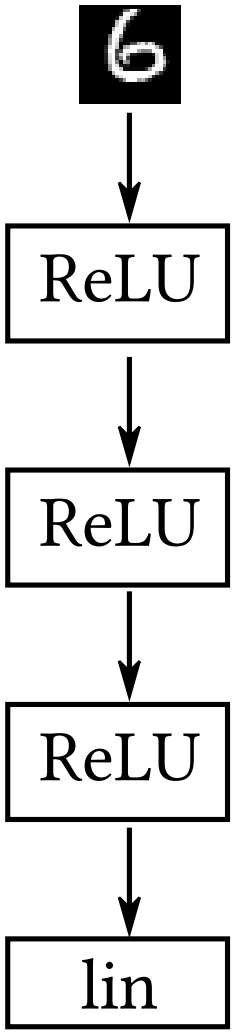
More layers





More layers

Rectified Linear Unit



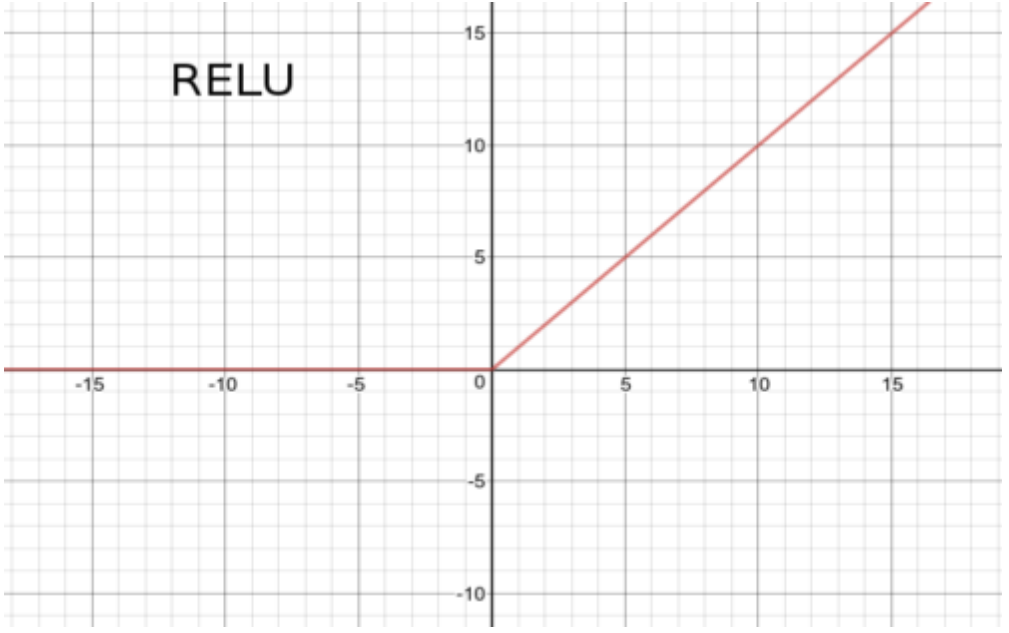
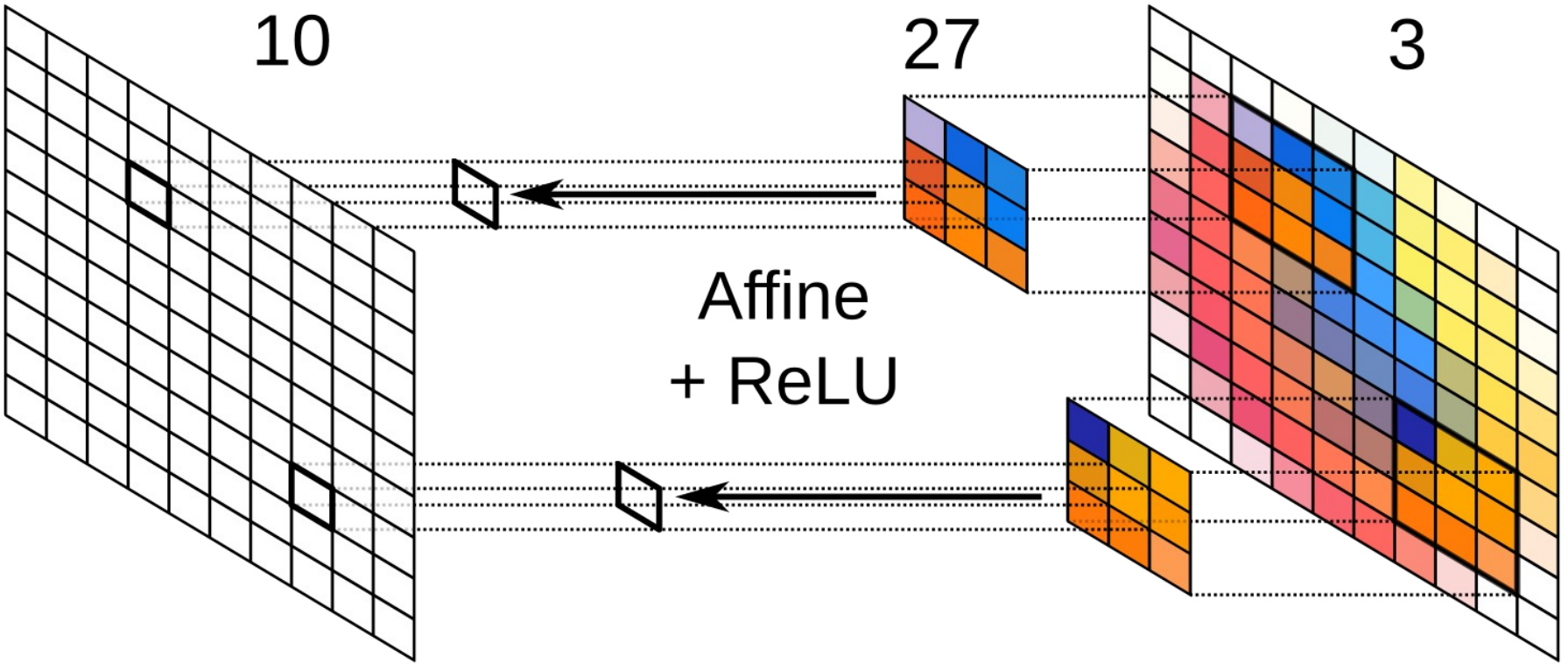


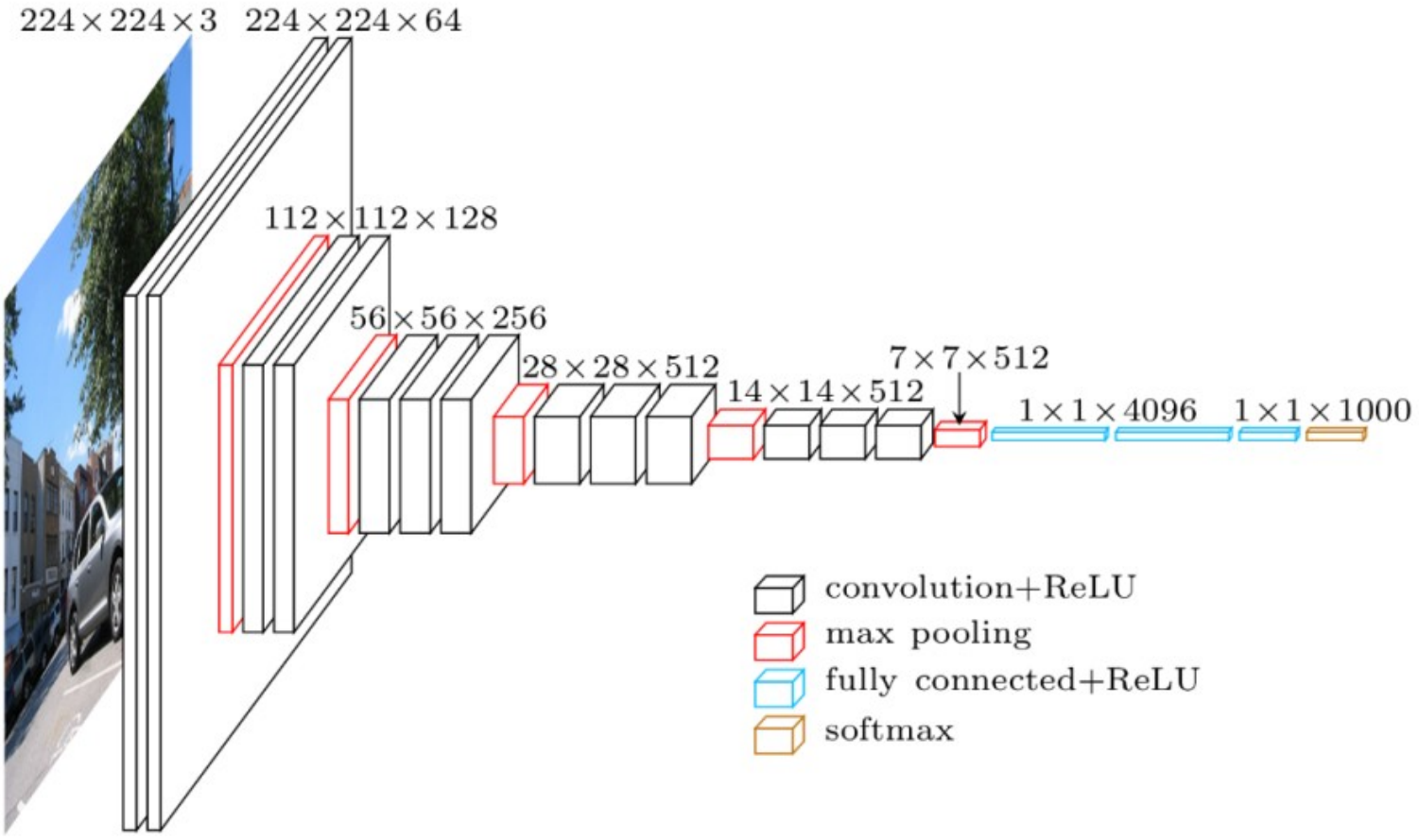
Image processing

- Objectives
  - Capture the grid structure of the image
  - Translation invariance

Convolutional layer



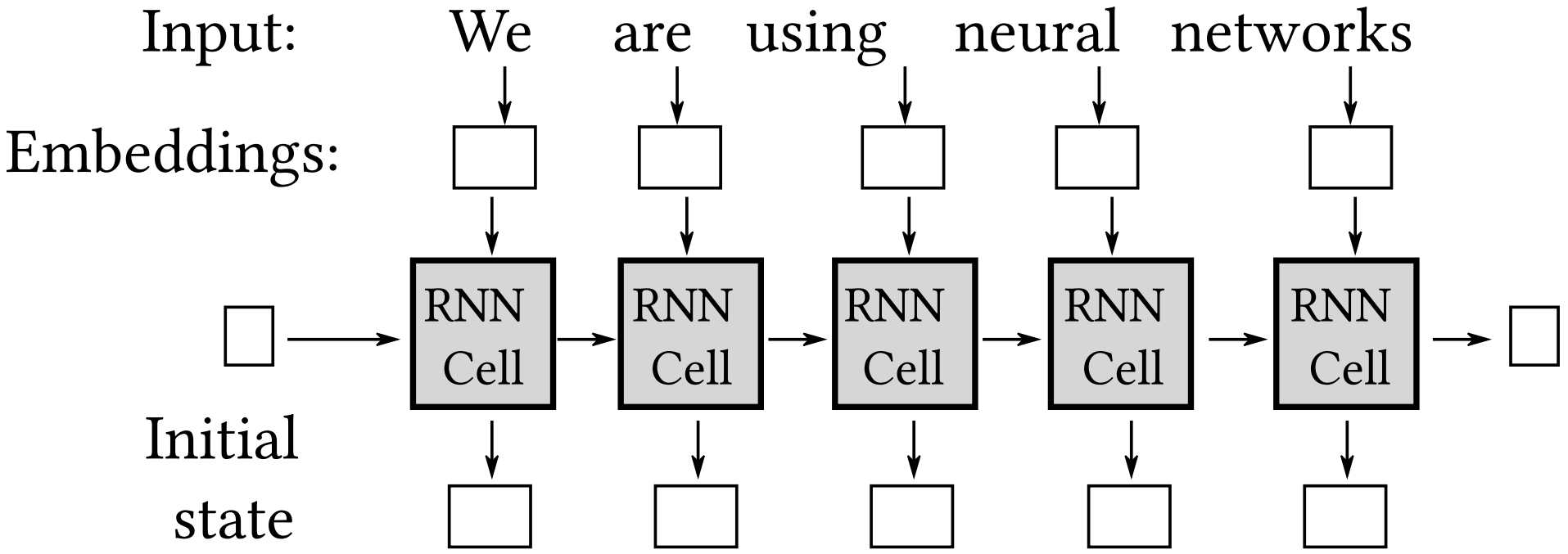
CNN Example


224×224×3
224×224×64
112×112×128
56×56×256
28×28×512
14×14×512
7×7×512
1×1×4096
1×1×1000
convolution+ReLU
max pooling
fully connected+ReLU
softmax

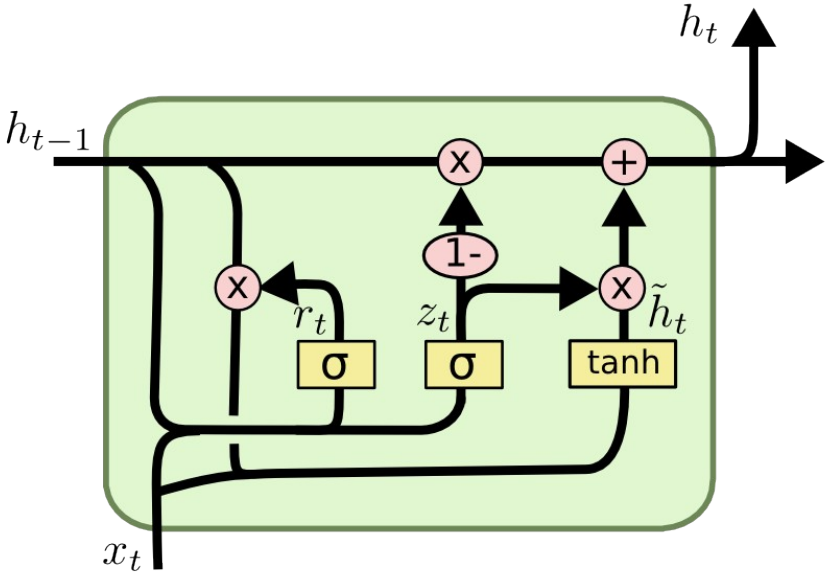
Natural language processing

- Step 1 – Word embedding
  - Dictionary: Word → learned real vector
- Step 2 – Recurrent Neural Network (RNN)
  - Objective: Handling sequences of variable lengths

Natural language processing



RNN Cell example



$$z_t = \sigma\left(W_z \cdot [h_{t-1}, x_t]\right)$$

$$r_t = \sigma\left(W_r \cdot [h_{t-1}, x_t]\right)$$

$$\tilde{h}_t = \tanh\left(W \cdot [r_t * h_{t-1}, x_t]\right)$$

$$h_t = (1 - z_t) * h_{t-1} + z_t * \tilde{h}_t$$

Summary

- Neural network = a differentiable function with many hidden parameters trained by stochastic gradient descent
- CNN for image recognition
- RNN for natural language