

Course

Foundations of Artificial Intelligence

Lecture 11 – Deep Learning 2

Dr. Mohsen Mesgar

Universität Duisburg-Essen

Deep Learning Architectures

Deep learning architectures

Processing images

- Convolutional neural networks

Processing sequences (e.g., text)

- Recurrent neural networks

Generating data

- Generative adversarial networks

Convolutional Neural Networks

Variable input in image recognition



The 1d convolution operation

$$(f * g)[i] = \sum_{m=-M}^{M} f[i - m]g[m]$$

- ◊ $*$ is the convolution operator
- ◊ $f(x)$ is the input representation
- ◊ i is the current position in the input representation
- ◊ M is the window size
- ◊ $g[m]$ is the weight for an input element with distance m to the current input
- ◊ → it is also often referred to as the filter.

- ◊ Careful! You will find many terminological alternatives in the literature: w (for weights) instead of g , n or t (for time) instead of i , a (for age) instead of m , ...

1-D Convolution

$$\begin{bmatrix} 1 & 1 & 1 & 0 & 0 \end{bmatrix} * \begin{bmatrix} 1 & 0 & 1 \end{bmatrix} = \begin{bmatrix} ? & ? & ? \end{bmatrix}$$

Input representation f Filter g Convolved output
(also ‘kernel’)

(Taken from the DL4NLP [course](#) 2021 (Ivan Habernal and Mohsen Mesgar)

1-D Convolution

$$\begin{bmatrix} \downarrow & & & & \\ 1 & 1 & 1 & 0 & 0 \end{bmatrix} * \begin{bmatrix} 1 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 2 & ? & ? \end{bmatrix}$$

Input representation f Filter g Convolved output

$$\begin{bmatrix} \downarrow & & & & \\ 1 & 1 & 1 & 0 & 0 \end{bmatrix} * \begin{bmatrix} 1 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 2 & 1 & ? \end{bmatrix}$$

Input representation f Filter g Convolved output

(Taken from the DL4NLP [course](#) 2021 (Ivan Habernal and Mohsen Mesgar)

2-D Convolution

$$\begin{bmatrix} 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \end{bmatrix} * \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 1 \end{bmatrix} = \begin{bmatrix} ? & ? & ? \\ ? & ? & ? \\ ? & ? & ? \\ ? & ? & ? \end{bmatrix}$$

Input representation f

Filter g
(also ‘kernel’)

Convolved output

(Taken from the DL4NLP [course](#) 2021 (Ivan Habernal and Mohsen Mesgar)

2-D Convolution

$$\begin{bmatrix} 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \end{bmatrix} * \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 4 & ? & ? \\ ? & ? & ? \\ ? & ? & ? \\ ? & ? & ? \end{bmatrix}$$

Filter g
(also 'kernel')

Input representation f Convolved output

$$\underbrace{1 \cdot 1 + 1 \cdot 0 + 1 \cdot 1}_{\text{1st row}} + \underbrace{0 \cdot 0 + 1 \cdot 1 + 1 \cdot 0}_{\text{2nd row}} + \underbrace{0 \cdot 1 + 0 \cdot 0 + 1 \cdot 1}_{\text{3rd row}} = 4$$

2-D Convolution

Move over all input regions

$$\begin{bmatrix} 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \end{bmatrix} * \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 4 & 3 & ? \\ ? & ? & ? \\ ? & ? & ? \\ ? & ? & ? \end{bmatrix}$$

Input representation f Filter g Convolved output
(also ‘kernel’)

2-D Convolution

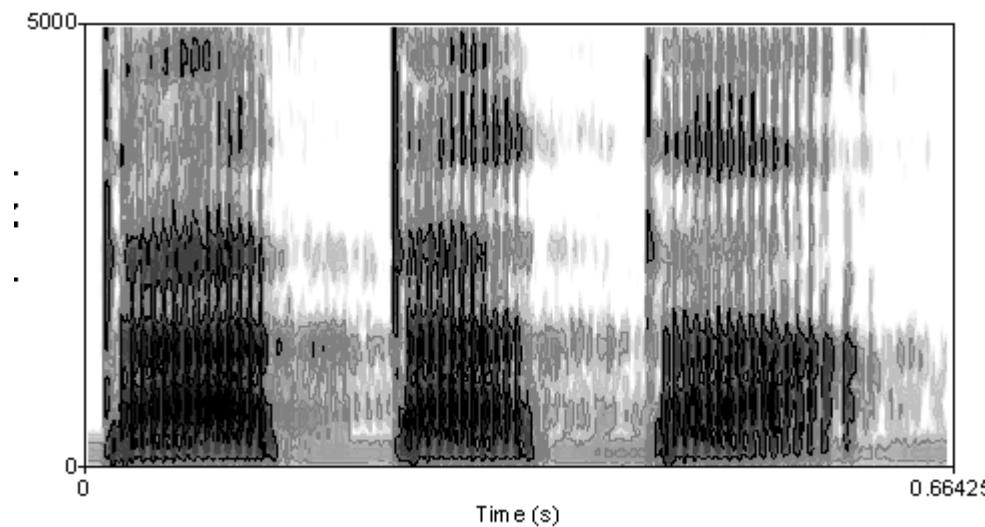
The goal is to learn good kernels (filter parameters)

$$\begin{bmatrix} 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & \textcolor{orange}{1} & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \end{bmatrix} * \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 4 & 3 & 4 \\ 2 & 4 & 3 \\ 2 & 3 & 4 \\ 2 & 3 & \textcolor{orange}{1} \end{bmatrix}$$

Input representation f Filter g Convolved output
(also ‘kernel’)

Convolutional networks in speech recognition

- ◊ Convolution over frequency to account for variations in the vocal tracts of different speakers



And in texts?

- ◊ Sentiment classification
- ◊ The movie was **really good**.
- ◊ We saw this **really good** movie.
- ◊ The movie, which we saw yesterday with all the colleagues in this tiny movie theatre next to the bridge, was (despite my expectations) **really good**.

- ◊ For this task, position information is not very relevant.

Dimensionality

- ◊ Advantages of text flow:
- ◊ Usually only one dimension
as opposed to two dimensions in images

Lore ipsum dolor sit amet, consectetuer adipiscing elit. Aenean ...



Convolution operation

- Input sentence $\mathbf{x}_{i:i+n}$ are stacked word embeddings of dimensionality d

- Convolutional filter $\mathbf{w} \in \mathbb{R}^{h \times d}$

where h is the filter size (how many neighboring words are convoluted)

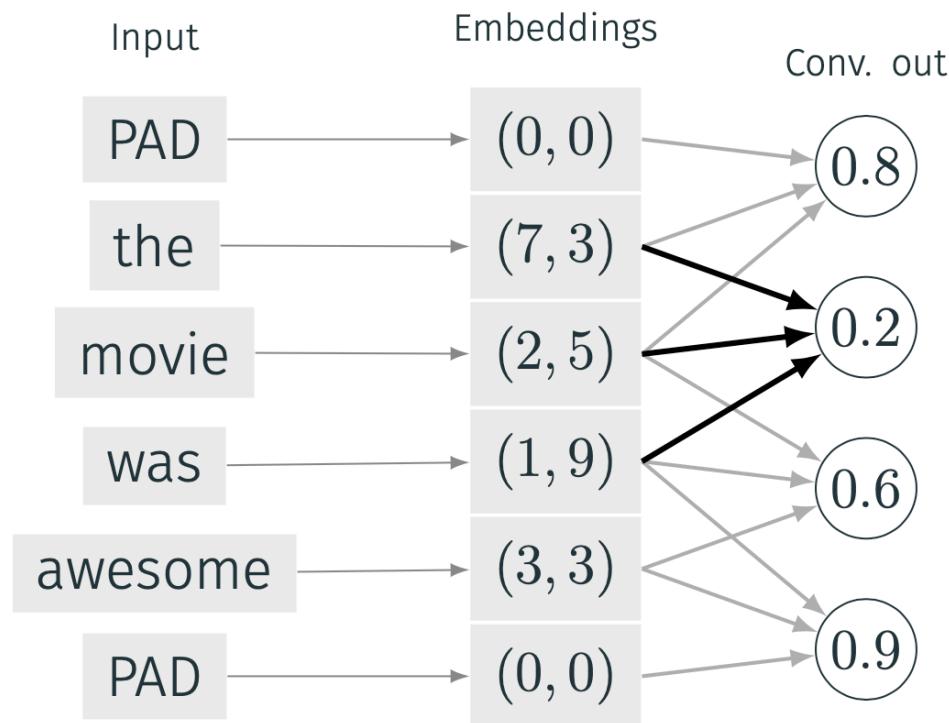
- Convolution operation $\mathbf{w} * \mathbf{x}_{i:i+n}$
- Non-linear operation

$$c_i = \text{ReLU}(\mathbf{w} * \mathbf{x}_{i:i+n} + b)$$

(Taken from the DL4NLP [course](#) 2021 (Ivan Habernal and Mohsen Mesgar)

Example

$$d = 2 \quad h = 3$$



(Taken from the DL4NLP [course](#) 2021 (Ivan Habernal and Mohsen Mesgar)

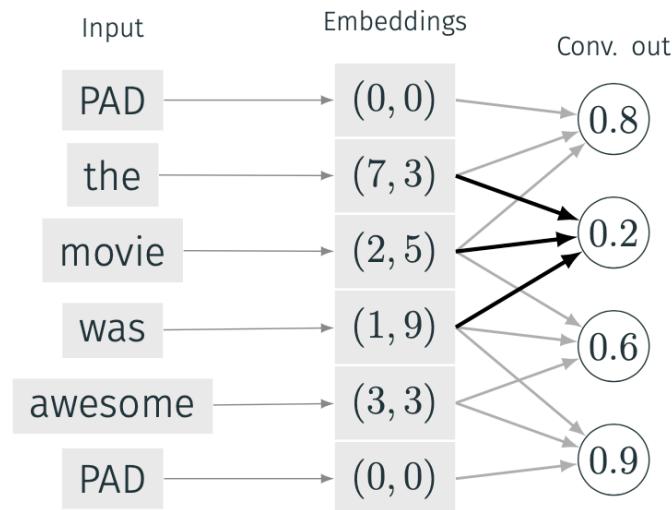
CNN vs FeedForward

Not every input is connected to every output in the following layer

- **sparse connectivity** (vs fully-connected/dense layers)

For each window, we use the same weights and bias values

- **parameter sharing**



Stride

Stride: the step size for moving over the sentence

- Stride 1 common in NLP; other values in comp. vision

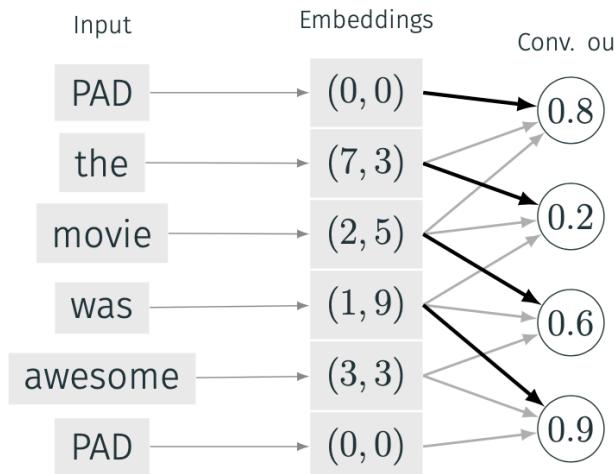


Figure 1: Stride = 1

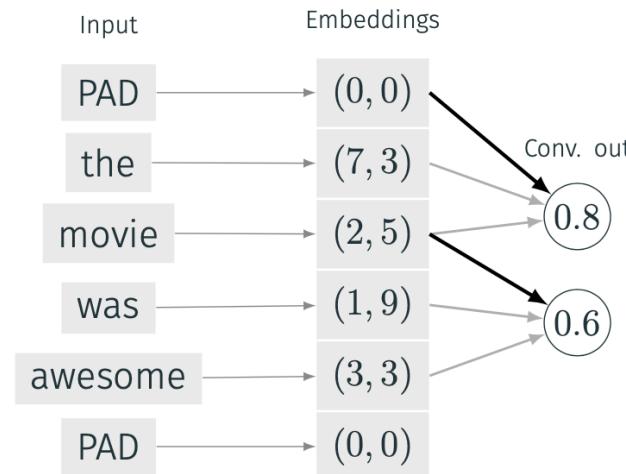


Figure 2: Stride = 2

(Taken from the DL4NLP [course](#) 2021 (Ivan Habernal and Mohsen Mesgar)

Pooling

Another new building block: pooling layer

- Idea: capture the most important activation

Let $c_1, c_2, \dots, \in \mathbb{R}$ denote the output values of the convolutional filter

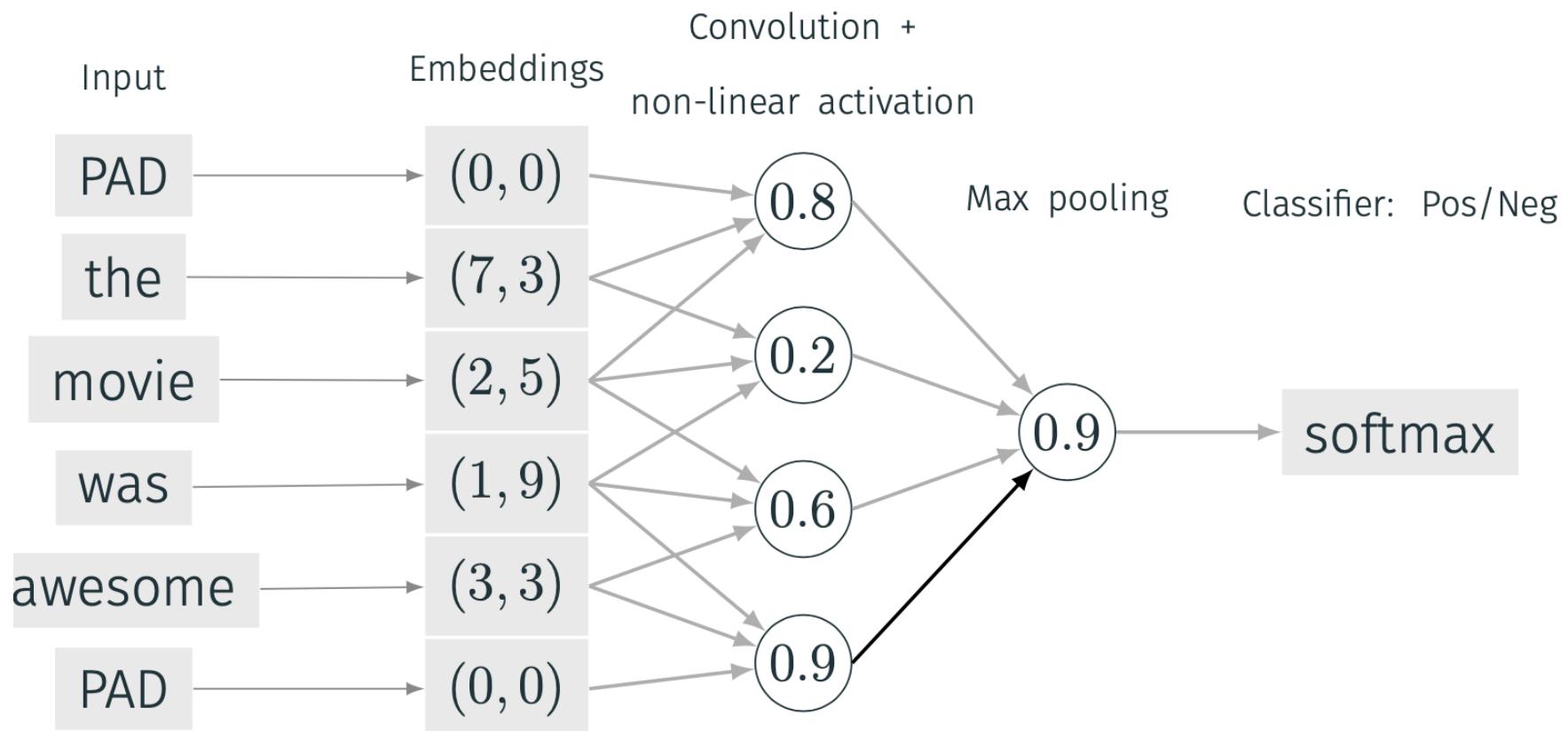
Compute output o for a **max-over-time pooling** layer as

$$o = \max_i c_i$$

Max-over-time pooling is most common in NLP. You can also find min-pooling and mean-pooling in other areas. Could also use some other averaging

Note that there are no associated weights

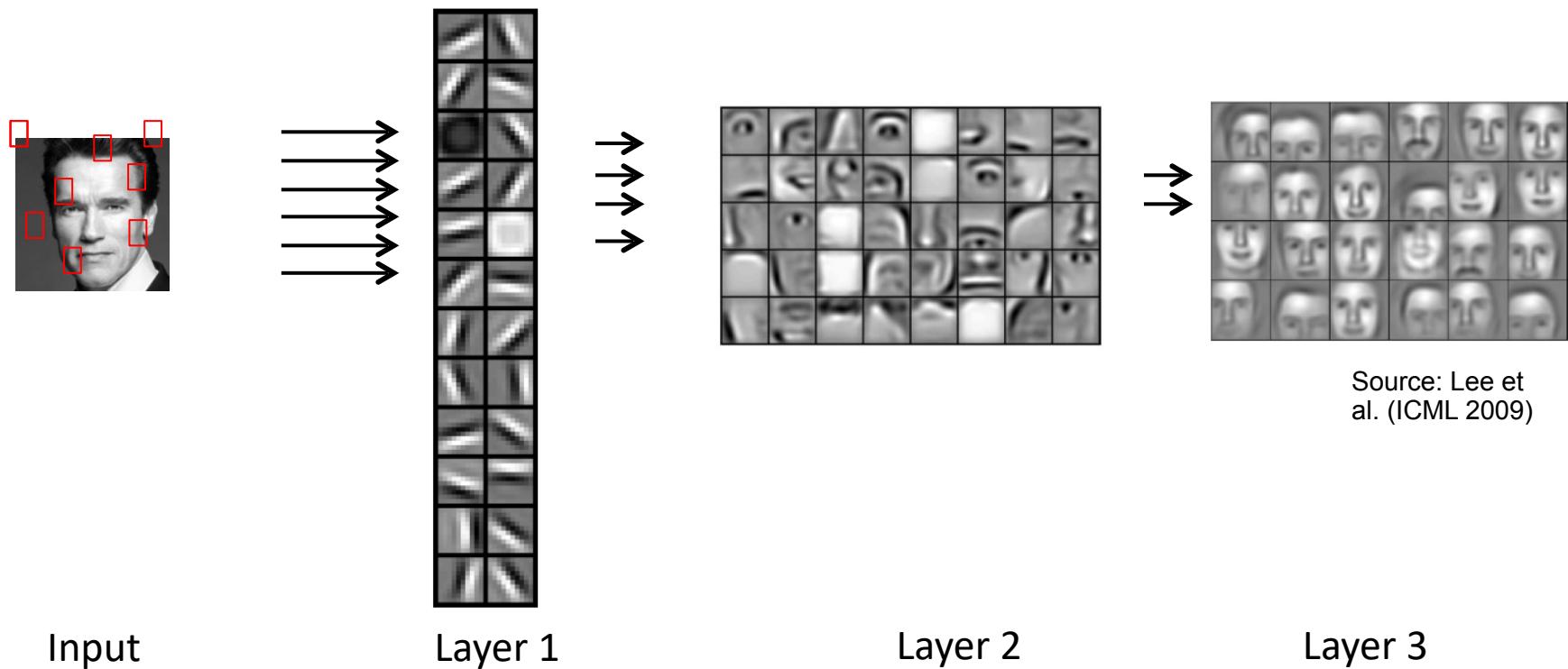
Pooling



(Taken from the DL4NLP [course](#) 2021 (Ivan Habernal and Mohsen Mesgar)

Stacked convolutional layers

- Convolutional layers can be stacked to derive high-level representations from simple representations
- In images: edge detection → identifying facial features (eyes, nose, lips) → face recognition



Stacked convolutional layers

- Convolutional layers can be stacked to derive high-level representations from simple representations
- In images: edge detection → identifying facial features (eyes, nose, lips) → face recognition
- In NLP?
- Polarity of words → Identifying polarity of phrase (negated positive phrase) → sentiment classification
- This is only a very rough approximation, usually features in neural networks are hard to interpret (if at all)

Recurrent Neural Networks

Representation of Language

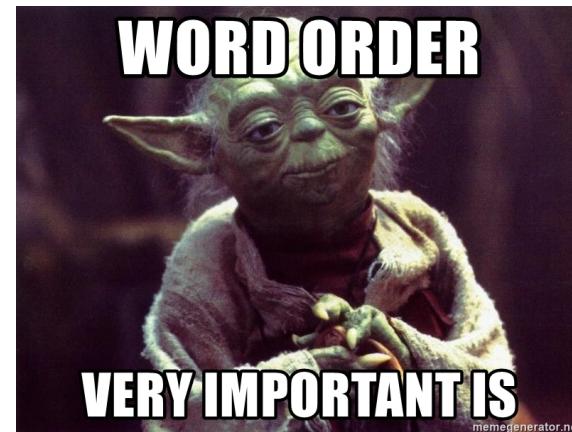
Neural networks need vectorized/numerical input

- This is an example sentence .
- [35, 5, 3, 134, 96, 10]

Fixed number of input neurons → fixed size

- [0 0 0 0 15 16 2 19 178 32]
- [52 154 462 33 89 78 285 16 145 95] 108 7 4 2
- [0 0 624 35 534 6 227 7 129 113]
- [26 49 2 15 566 30 579 21 64 2]
- [19 14 5 2 6 226 251 7 61 113] 4 226 65 12
10 10

Word order matters ...



„Man bites dog“
„Dog bites man“



Long-range Dependencies

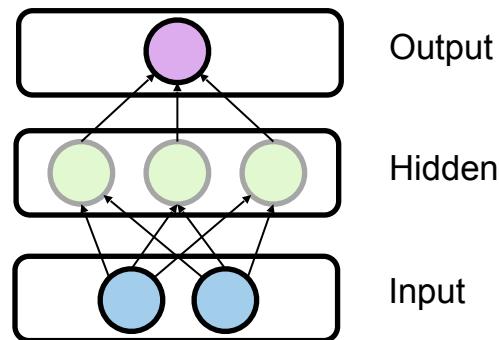
e.g. Sentiment classification

- I like the new iPhone but it doesn't have any new features and it is much too expensive.

Long-range Dependencies

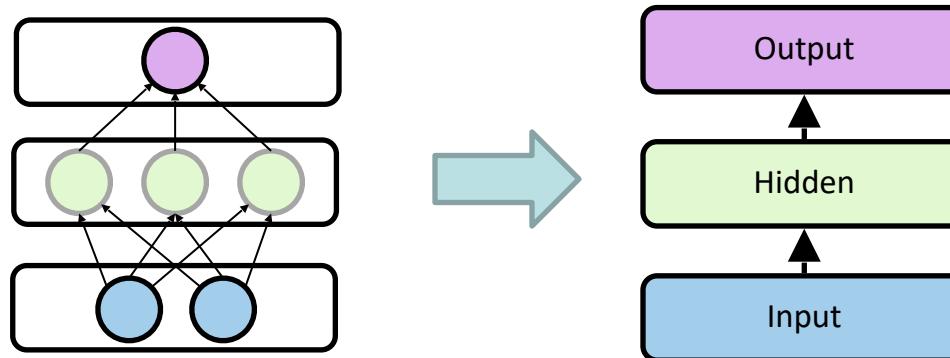
e.g. Sentiment classification

- [I like the new iPhone but it doesn't have any new features and it is] ~~much too expensive.~~



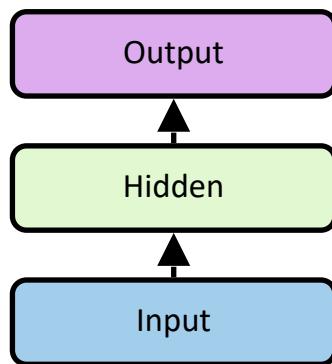
How to model long input sequences?

Collapsed View

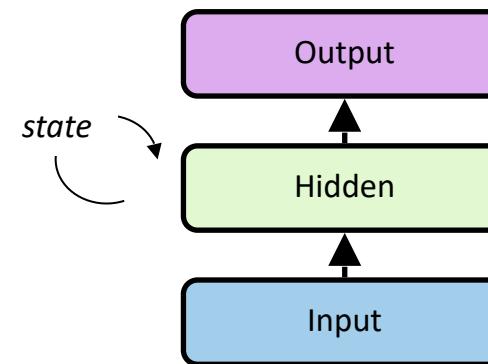


From Feed-forward to Recurrent Nets

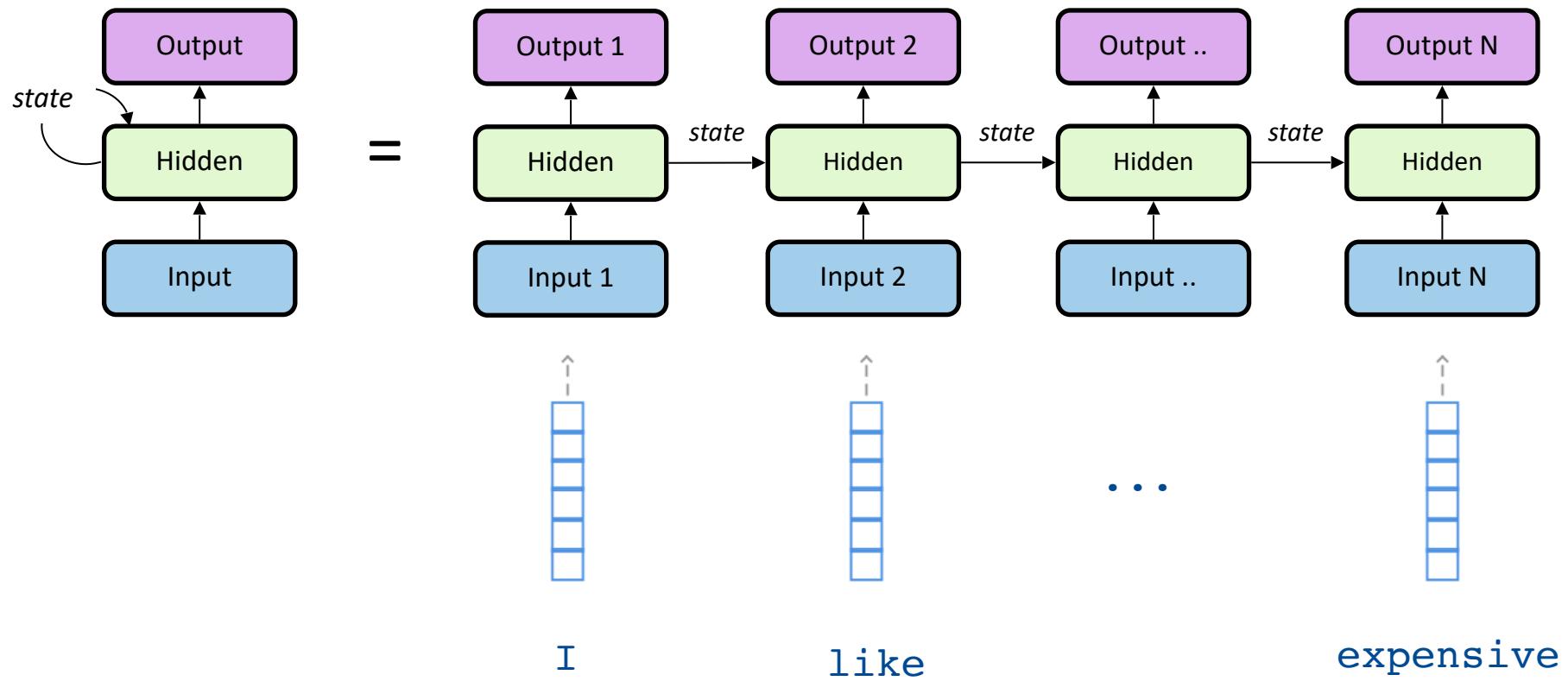
Feed-forward



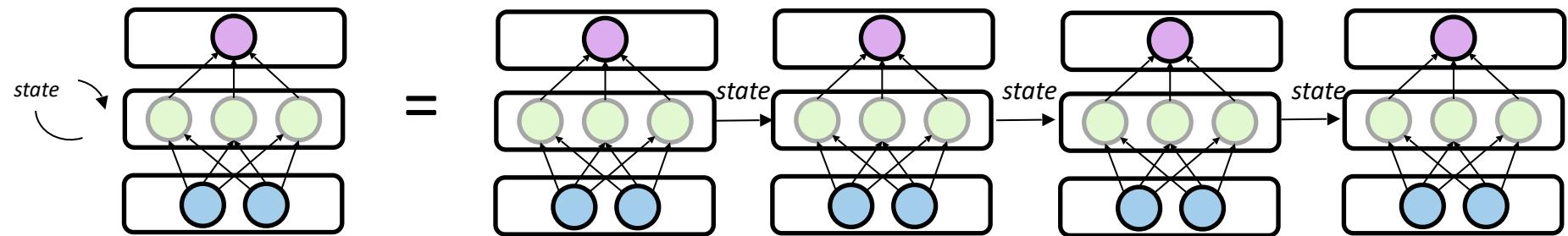
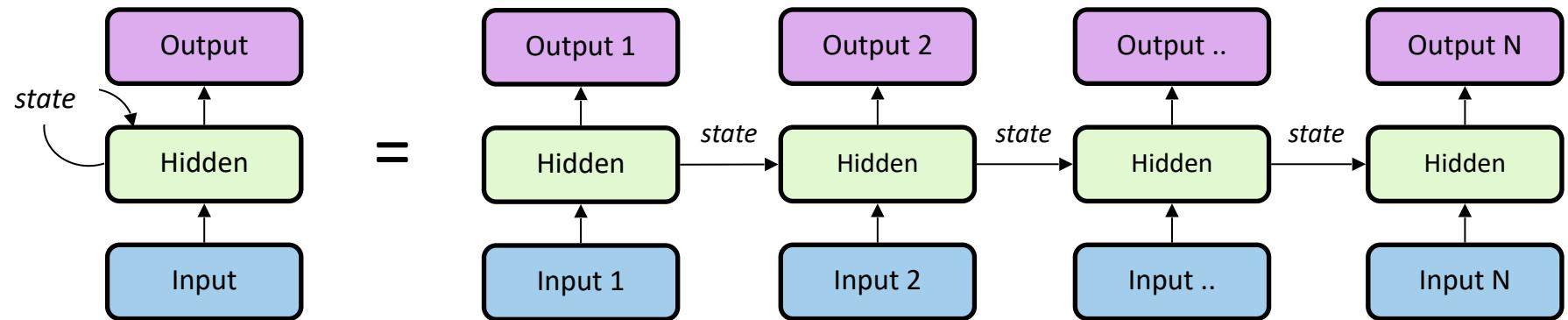
Recurrent



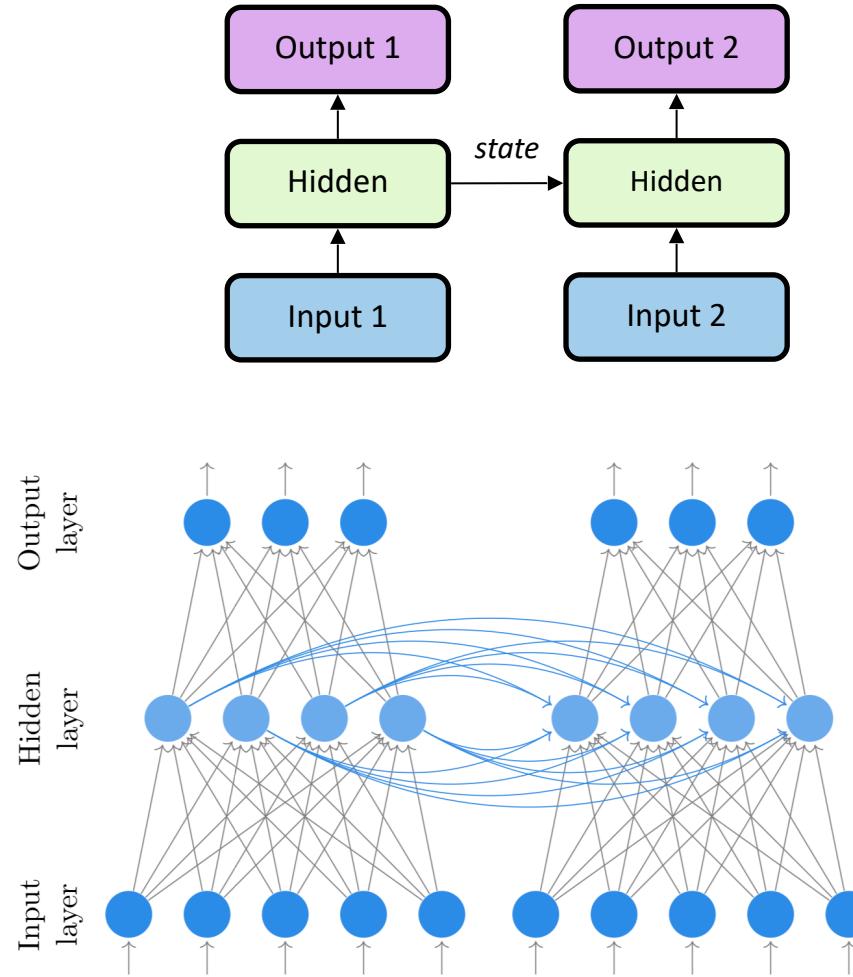
Unrolling in time



Unrolling in time

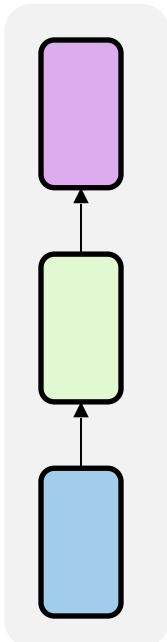


Zoomed-in View



Variants

one to one



Vanilla Feed-Forward

Variants

one to one one to many

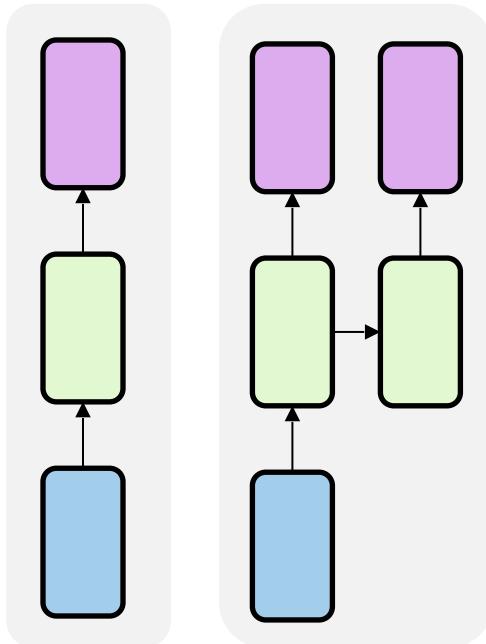
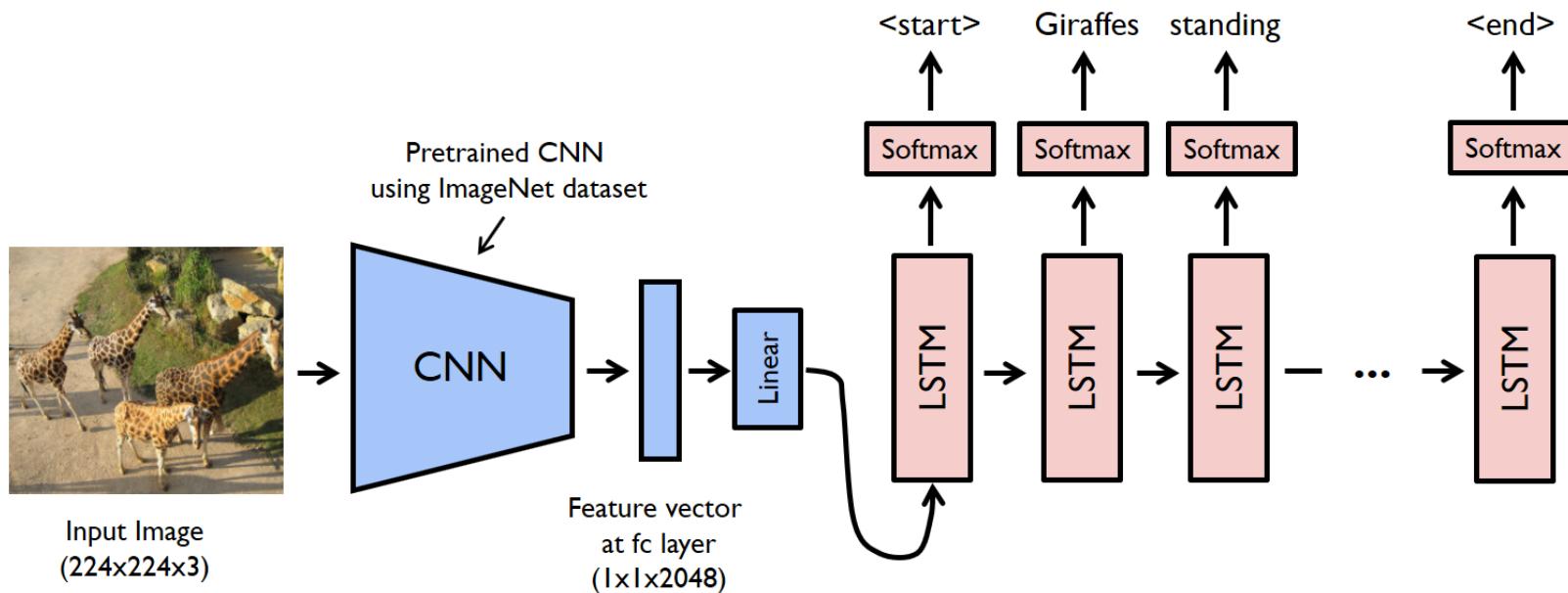


Image Captioning

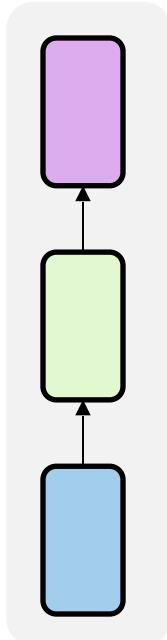
- *one image to many words*

Image Captioning

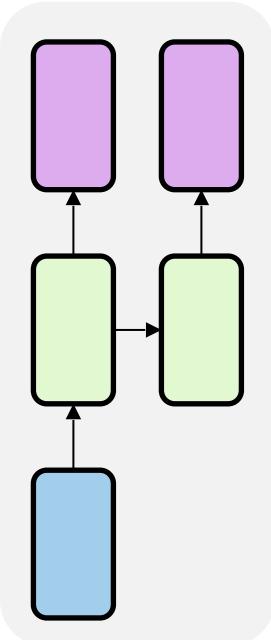


Variants

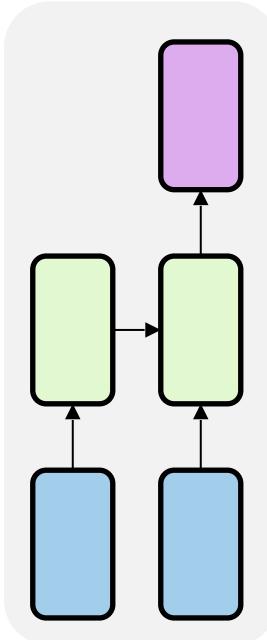
one to one



one to many



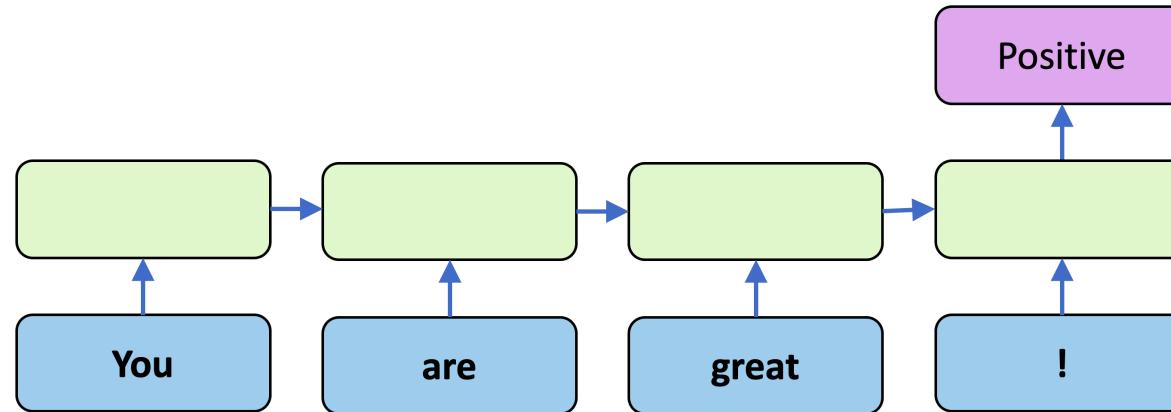
many to one



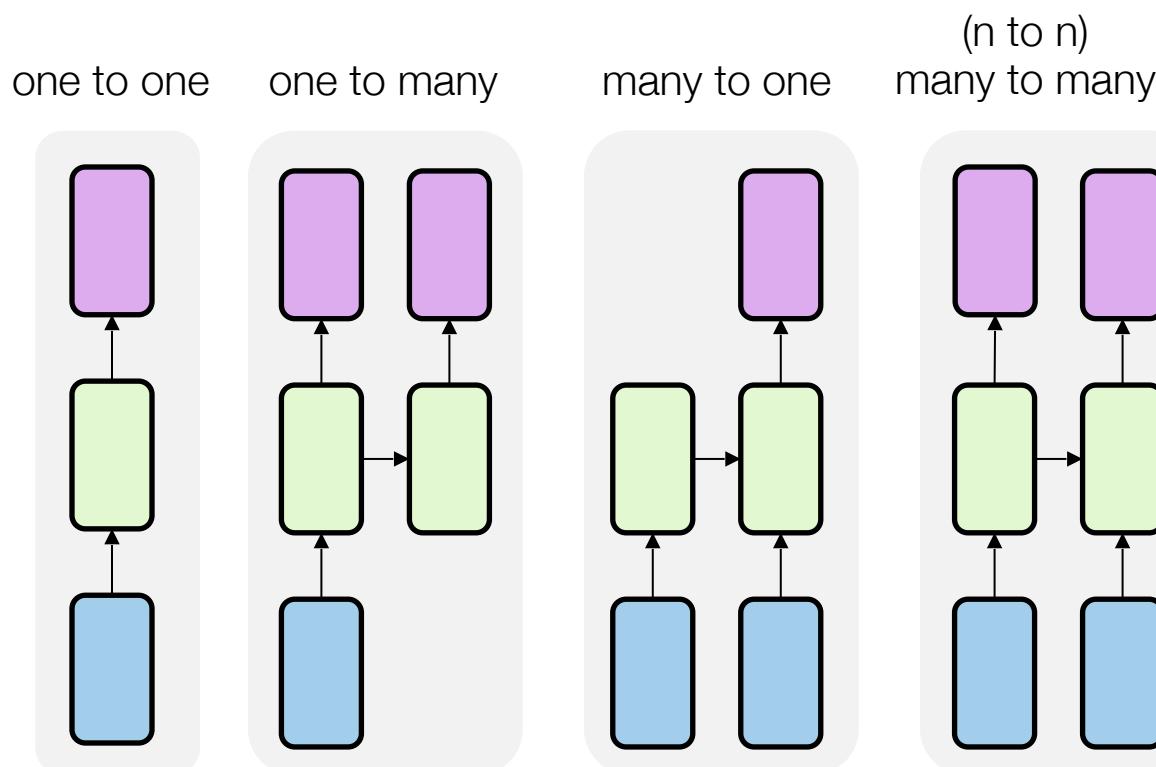
Sentiment Classification

- *many words to one category*

Sentiment Classification



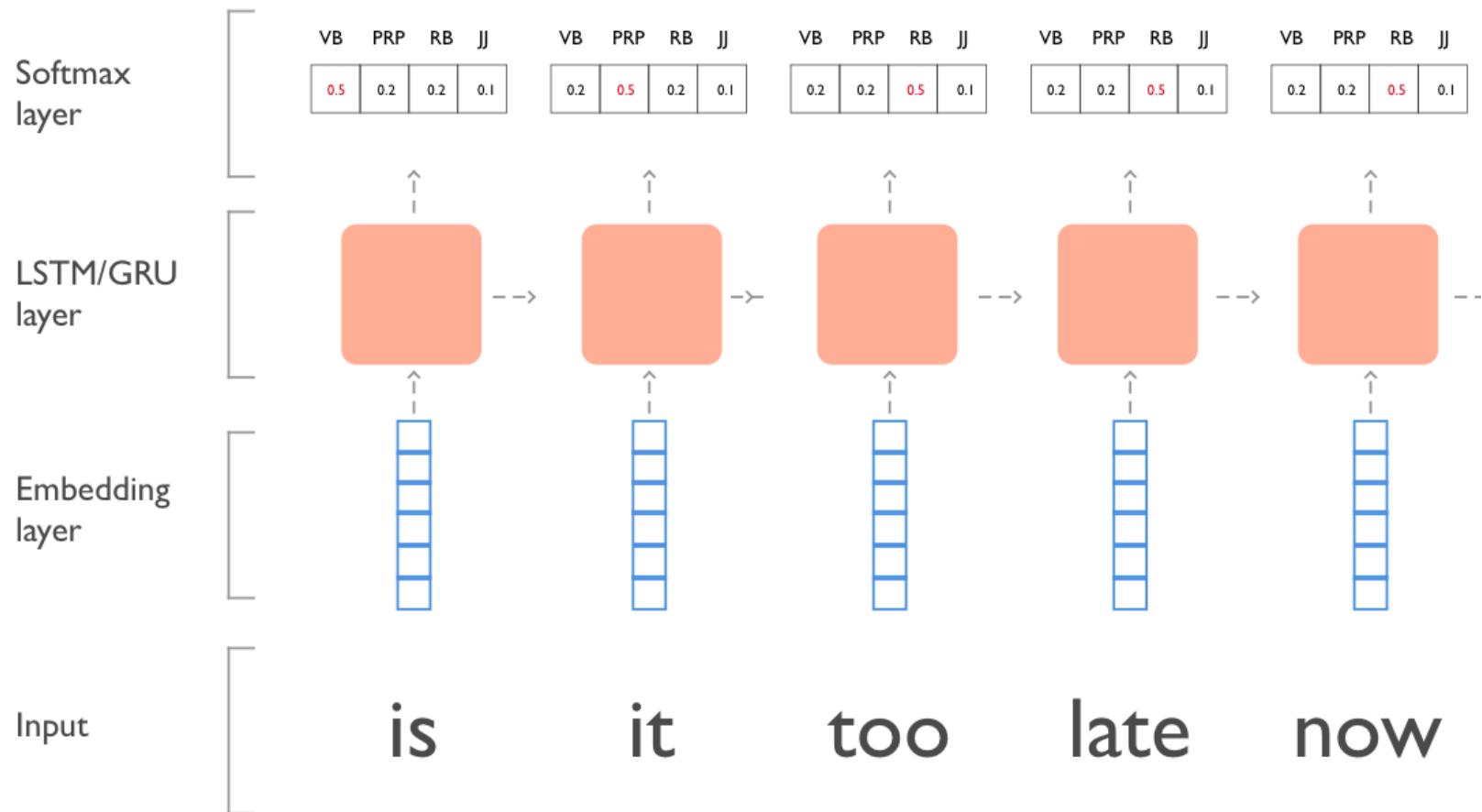
Variants



POS Tagging

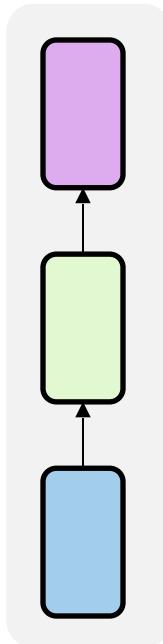
- *many words to many tags*

POS Tagging

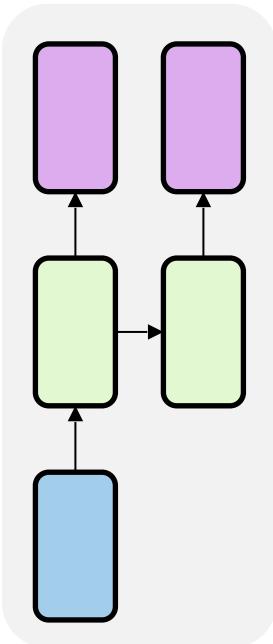


Variants

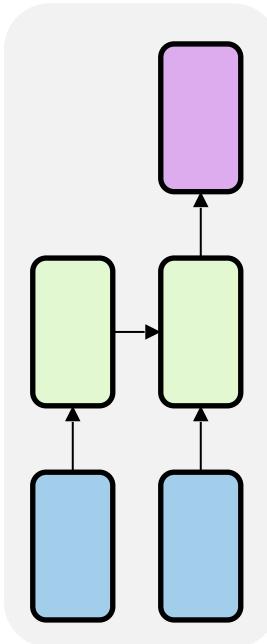
one to one



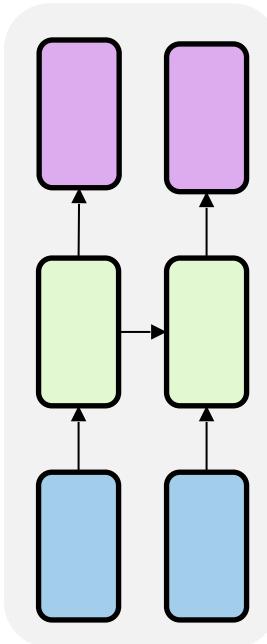
one to many



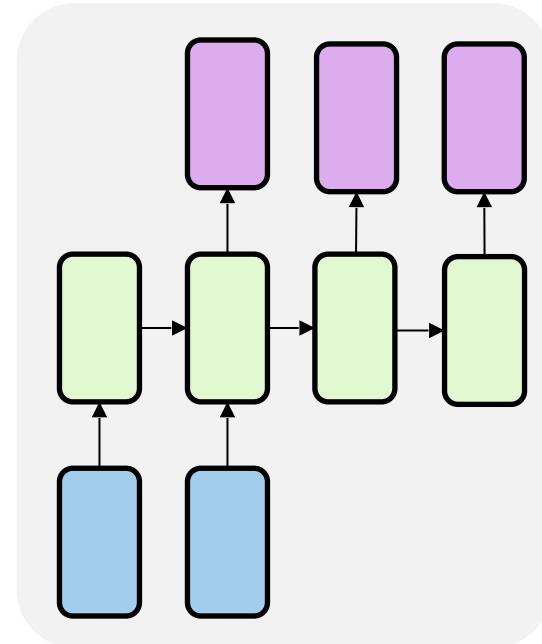
many to one



(n to n)
many to many



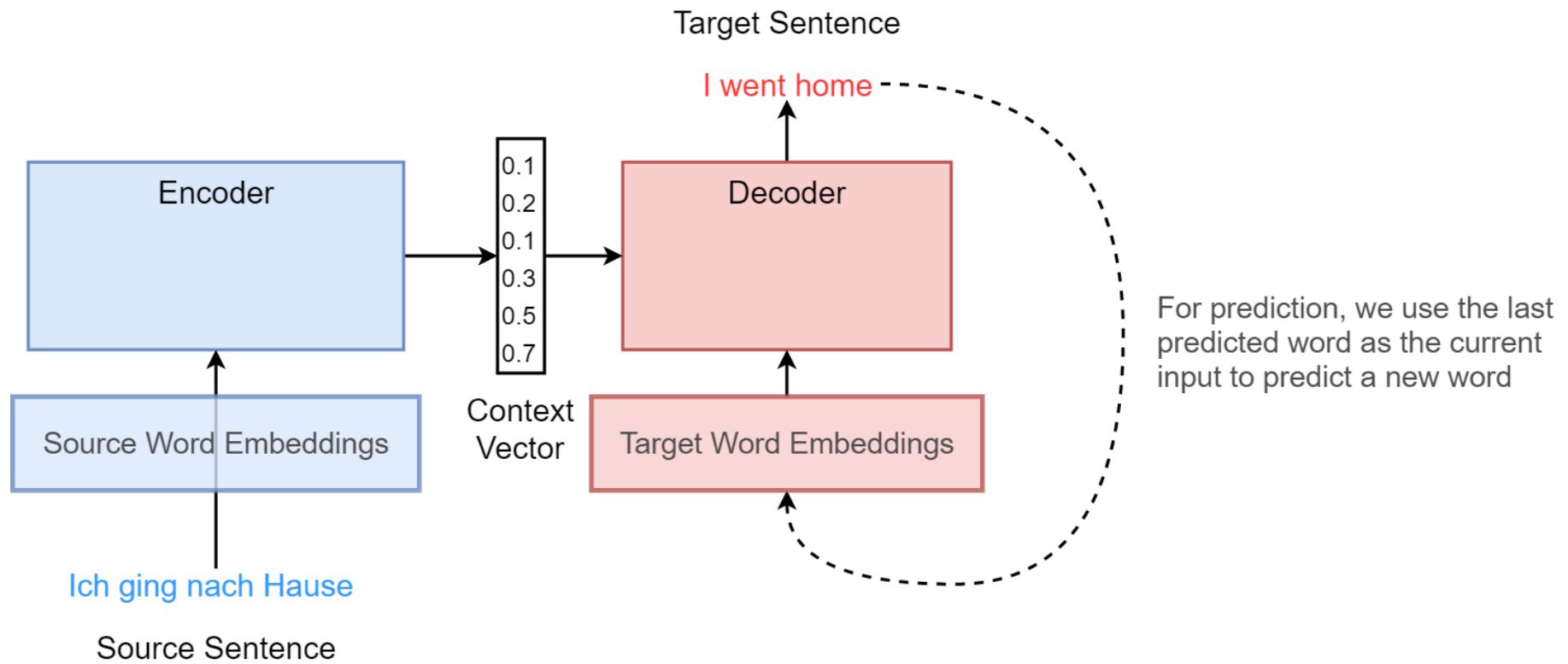
(n to m)
many to many



Machine Translation

- *many words to many words*

Machine Translation



Generative Adversarial Networks (GANs)

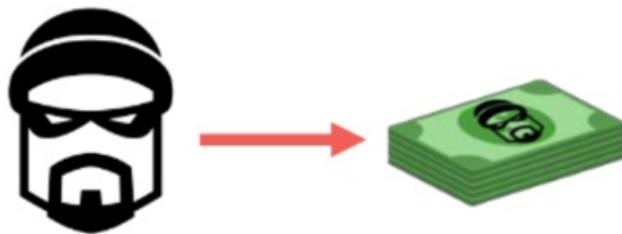
Generative networks

- The networks we have seen so far, read an input matrix of data and modify it by using weights and activation functions.
- In the final layer, we run a softmax over the modified matrix to receive classification probabilities.
- The input matrix represents the input data.
- Could we modify the matrix such that it represents new data?
→ generative model
- For many tasks, we encounter the problem that we do not have enough training data.
Couldn't we just generate it?

Generator-discriminator competition

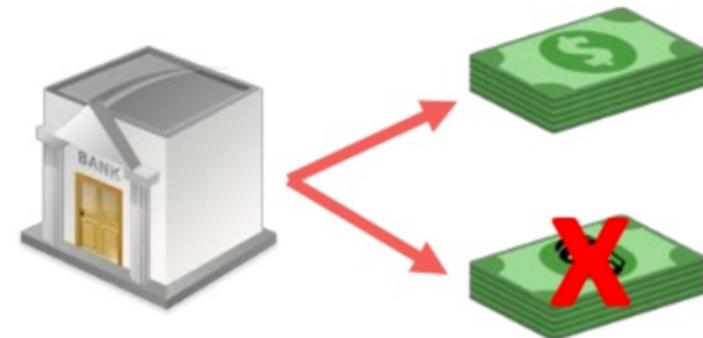
- ◊ Generator: try to generate authentic looking instances.
- ◊ Discriminator: distinguish between the real and generated instances.

Generator



Goal: produce counterfeit money that is as similar as real money.

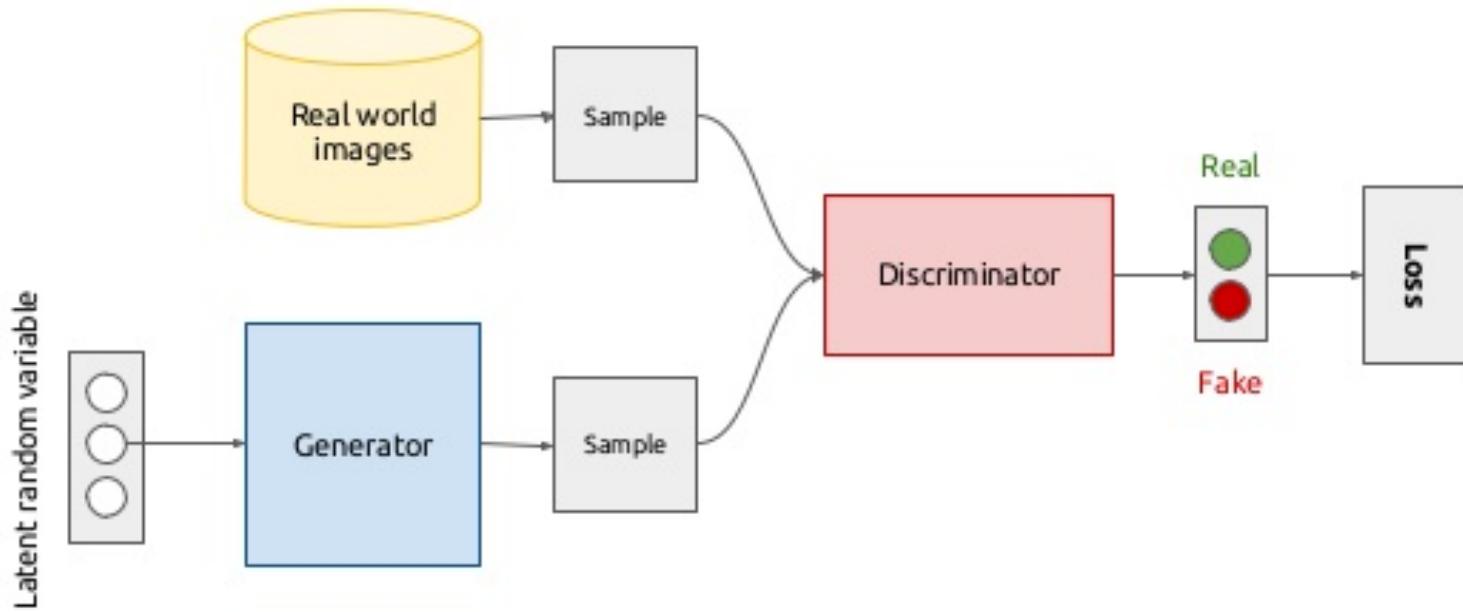
Discriminator (Classifier)



Goal: distinguish between real and counterfeit money.

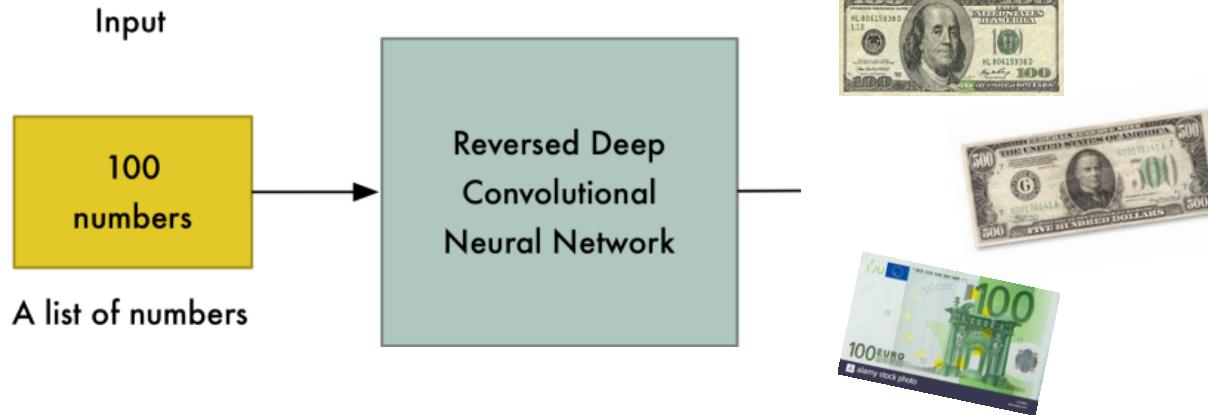
Generative Adversarial Networks (GAN)

- During training, both the discriminator and the generator improve.

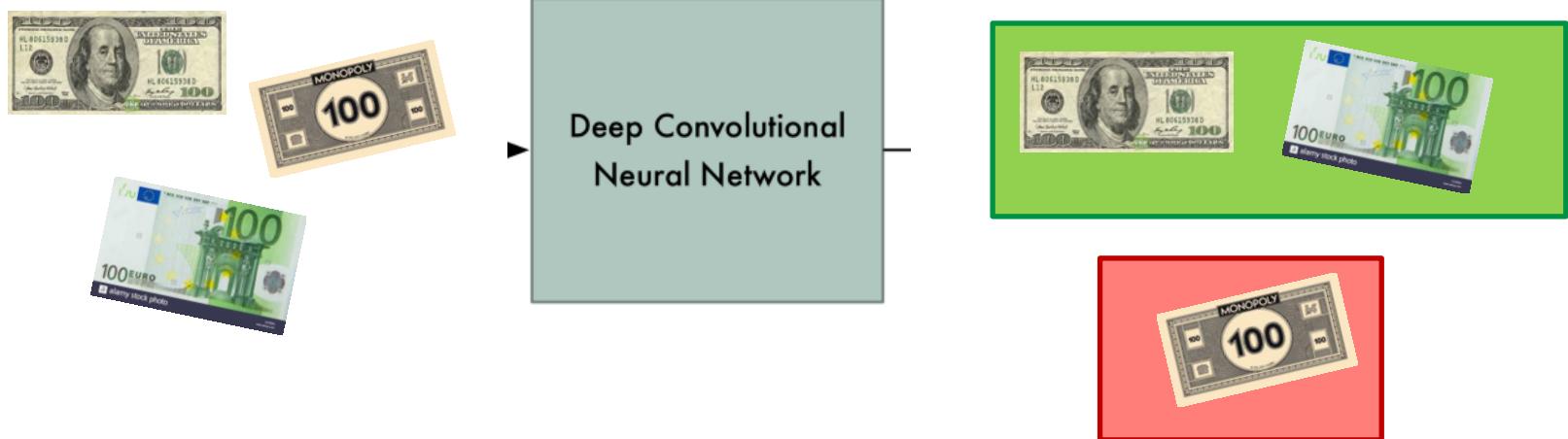


Objective

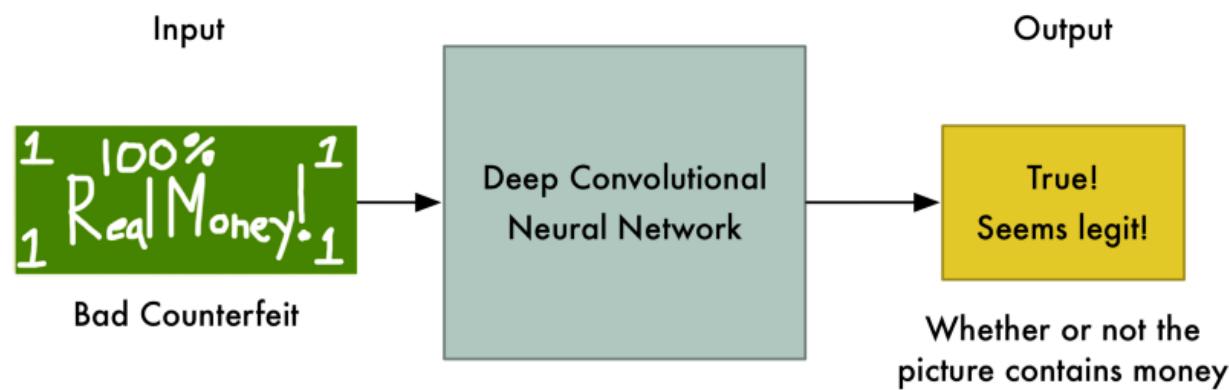
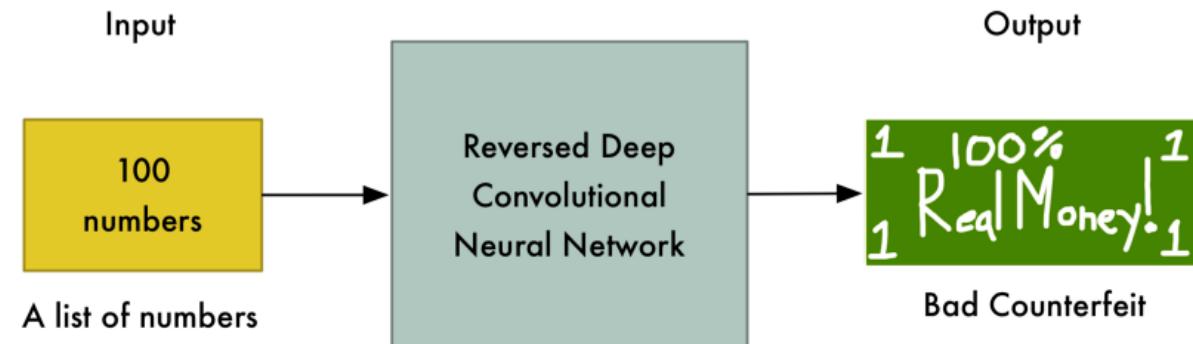
Generator:



Discriminator:

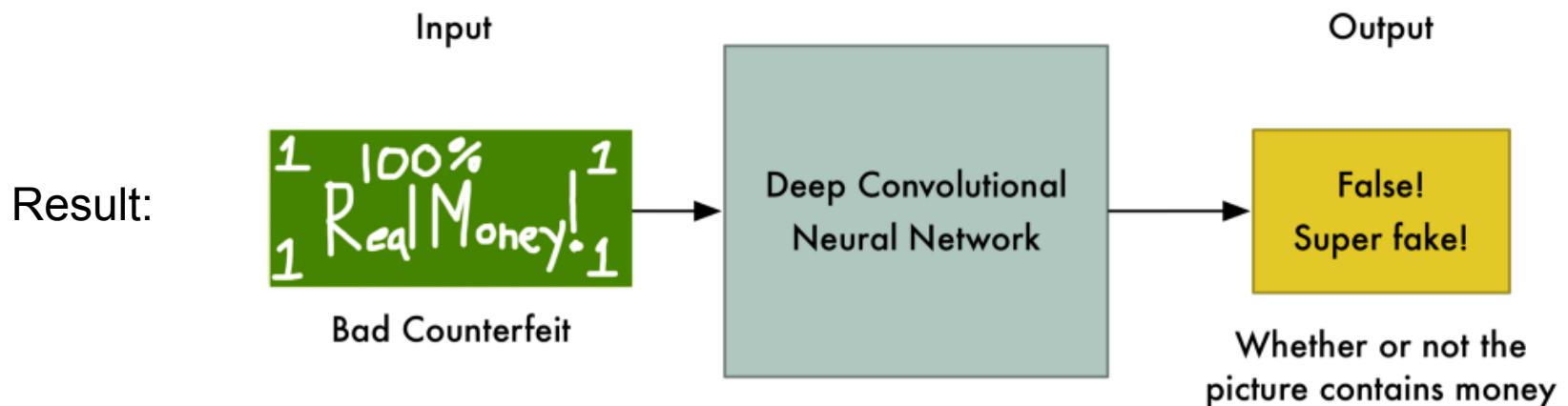


GAN: Start



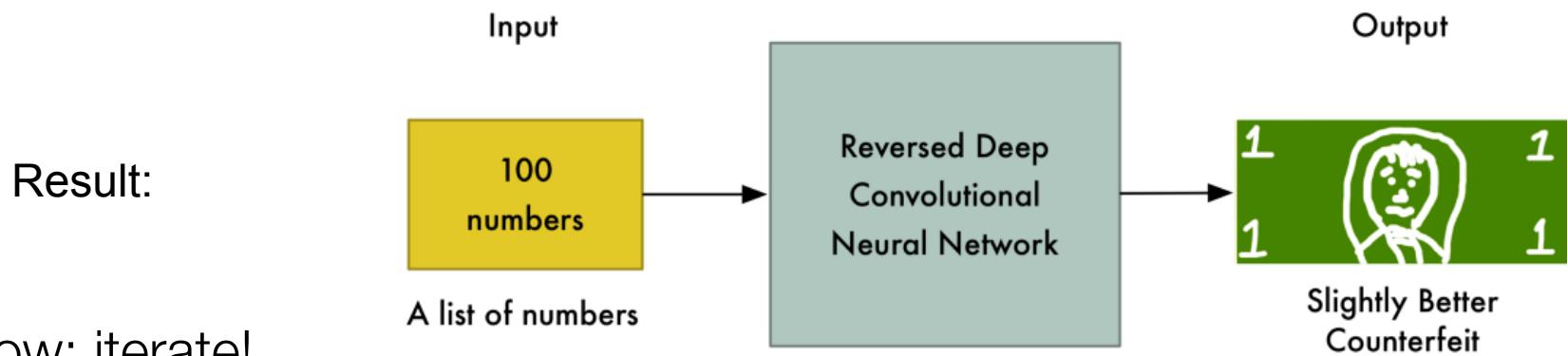
GAN: Improve classification

- The discriminator learns from sample data to distinguish between real and fake money.
 - e.g. real money has a picture of a person on it



GAN: Improve generation

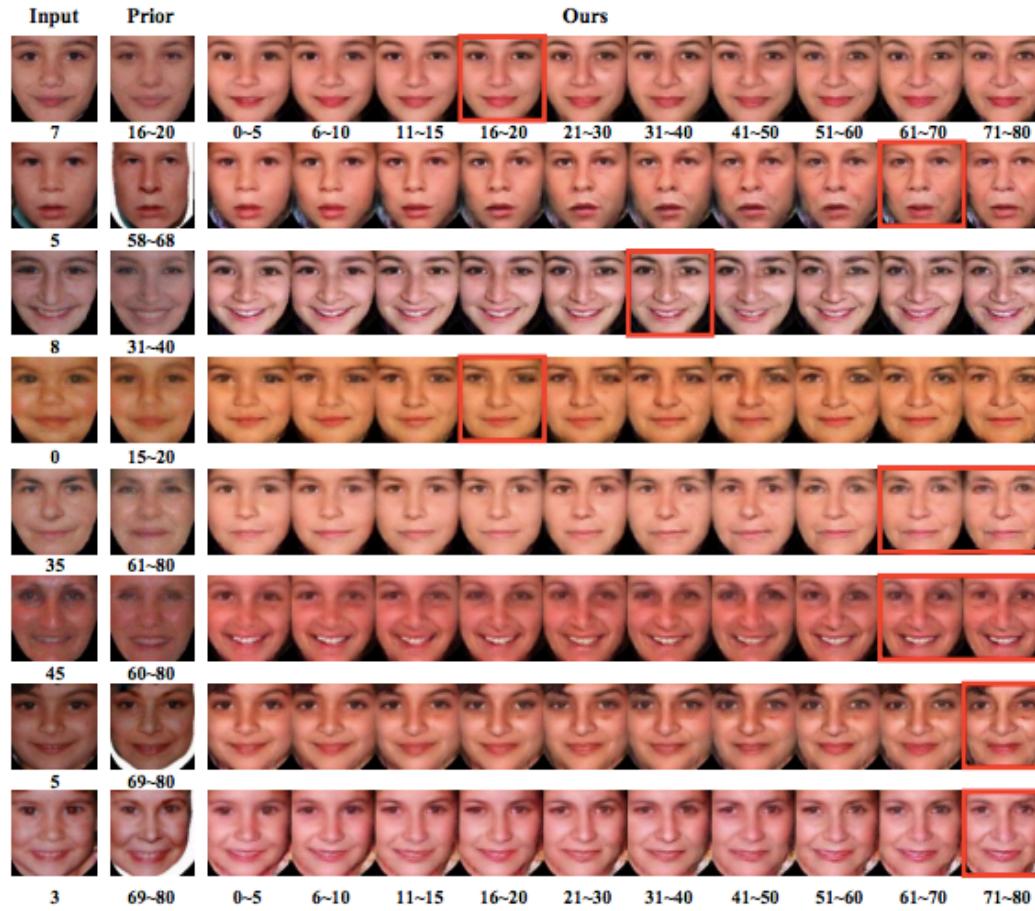
- The generator learns that the generated samples are getting rejected.
- It learns which features the discriminator is looking for (pictures of people) and improves the generation.



- Now: iterate!
- An improved generator leads to an improved discriminator leads to an improved generator...

Face aging

Zhang et al: <https://arxiv.org/pdf/1702.08423.pdf>



Generating art

Elgammal et al.: <https://arxiv.org/pdf/1706.07068.pdf>



Summary

- Different architectures serve different purposes:
- Convolutional neural networks
 - Image processing, document classification
- Recurrent neural network
 - Sequence processing
- Generative adversarial neural networks
 - Artificial data generation

Reading

Mandatory:

- Ian Goodfellow, Yoshua Bengio, Aaron Courville:
- Deep Learning, *Chapter 9*, 9.1 to 9.3.
- Deep Learning, *Chapter 10*, 10.2 and 10.4

<http://www.deeplearningbook.org/>

Optional:

- <http://colah.github.io/posts/2015-08-Understanding-LSTMs/>
- <https://machinelearningmastery.com/what-are-generative-adversarial-networks-gans/>