

Hochschule für Technik  
und Wirtschaft Berlin

*University of Applied Sciences*

# Deep Learning Intro

# Illustration

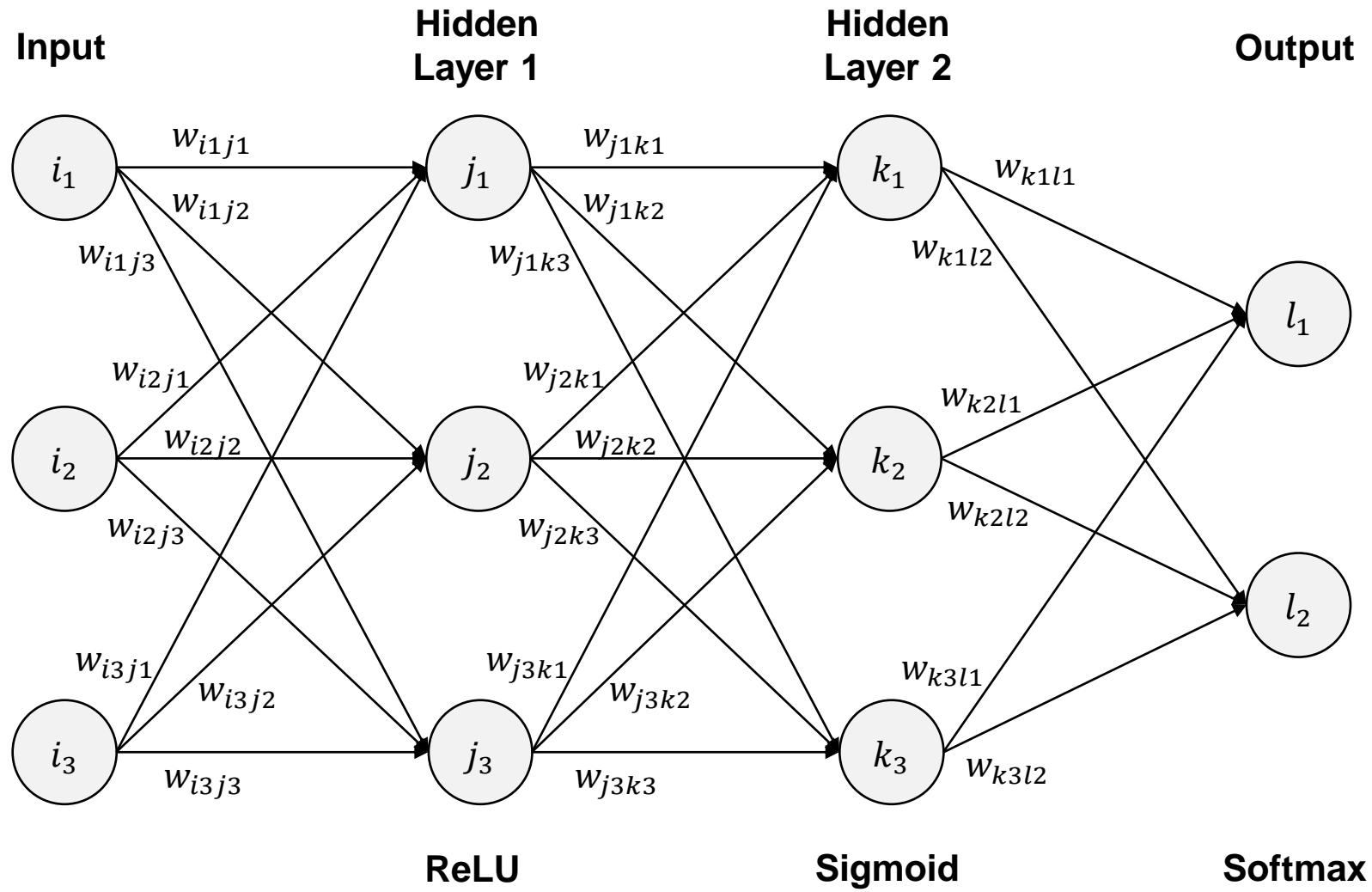
## ■ Approach

- Build a small neural network as defined in the architecture below.
- Initialize the weights and bias randomly.
- Fix the input and output.
- Forward pass the inputs and calculate the cost.
- Compute the gradients and errors.
- Backprop and adjust the weights and bias accordingly

## ■ Architecture:

- Build a Feed Forward neural network with 2 hidden layers. Hidden layers will have 3 neurons each, output layer 2 neurons.
- 1st and 2nd hidden layer will have Relu and Sigmoid respectively as activation functions. Final layer will have Softmax.
- Error is calculated using mean squared error.

# Feed Forward Neural Network



# Initialization

Initialization:

$$W_{ij} = \begin{bmatrix} w_{i1j1} & w_{i1j2} & w_{i1j3} \\ w_{i2j1} & w_{i2j2} & w_{i2j3} \\ w_{i3j1} & w_{i3j2} & w_{i3j3} \end{bmatrix} = \begin{bmatrix} 0.1 & 0.2 & 0.3 \\ 0.3 & 0.2 & 0.7 \\ 0.4 & 0.3 & 0.9 \end{bmatrix}$$

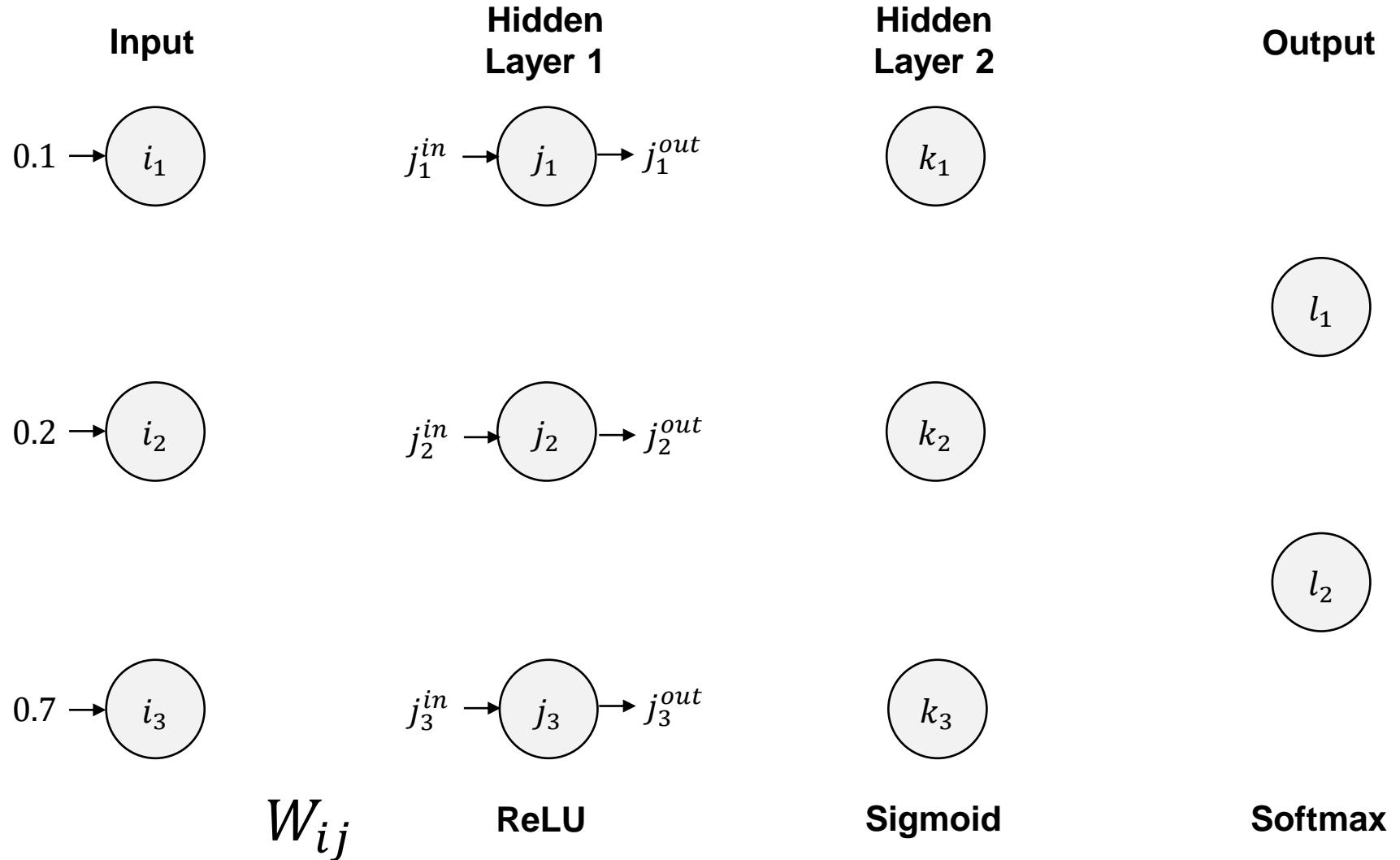
$$W_{jk} = \begin{bmatrix} w_{j1k1} & w_{j1k2} & w_{j1k3} \\ w_{j2k1} & w_{j2k2} & w_{j2k3} \\ w_{j3k1} & w_{j3k2} & w_{j3k3} \end{bmatrix} = \begin{bmatrix} 0.2 & 0.3 & 0.5 \\ 0.3 & 0.5 & 0.7 \\ 0.6 & 0.4 & 0.8 \end{bmatrix}$$

$$W_{kl} = \begin{bmatrix} w_{k1l1} & w_{k1l2} \\ w_{k2l1} & w_{k2l2} \\ w_{k3l1} & w_{k3l2} \end{bmatrix} = \begin{bmatrix} 0.1 & 0.4 \\ 0.3 & 0.7 \\ 0.5 & 0.2 \end{bmatrix}$$

Data:

Record ID	$i_1$	$i_2$	$i_3$	$l_1$	$l_2$
1	0.1	0.2	0.7	1.0	0.0
...	...	...	...	...	...
n	0.5	0.3	0.9	0.0	1.0

# Layer 1



# Layer 1 – Matrix Operation

$$\mathbf{i} * W_{ij} + \mathbf{b}_j = \mathbf{j}^{in}$$

$$[i_1 \quad i_2 \quad i_3] * \begin{bmatrix} w_{i1j1} & w_{i1j2} & w_{i1j3} \\ w_{i2j1} & w_{i2j2} & w_{i2j3} \\ w_{i3j1} & w_{i3j2} & w_{i3j3} \end{bmatrix} + [b_{j1} \quad b_{j2} \quad b_{j3}] = [j_1^{in} \quad j_2^{in} \quad j_3^{in}]$$

$$\begin{bmatrix} i_1 * w_{i1j1} + i_2 * w_{i2j1} + i_3 * w_{i3j1} \\ i_1 * w_{i1j2} + i_2 * w_{i2j2} + i_3 * w_{i3j2} \\ i_1 * w_{i1j3} + i_2 * w_{i2j3} + i_3 * w_{i3j3} \end{bmatrix}^T + [b_{j1} \quad b_{j2} \quad b_{j3}] = [j_1^{in} \quad j_2^{in} \quad j_3^{in}]$$

**Example:**

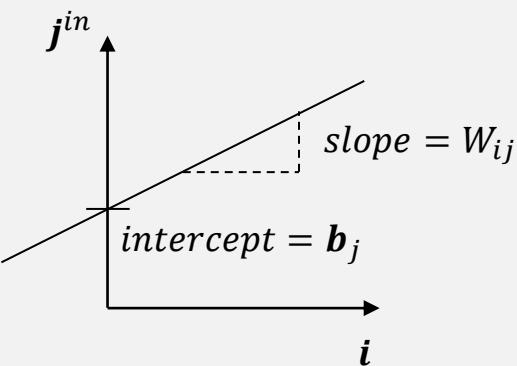
$$[0.1 \quad 0.2 \quad 0.7] * \begin{bmatrix} 0.1 & 0.2 & 0.3 \\ 0.3 & 0.2 & 0.7 \\ 0.4 & 0.3 & 0.9 \end{bmatrix} + [1.0 \quad 1.0 \quad 1.0]$$

$$= \begin{bmatrix} 0.1 * 0.1 + 0.2 * 0.3 + 0.7 * 0.4 \\ 0.1 * 0.2 + 0.2 * 0.2 + 0.7 * 0.3 \\ 0.1 * 0.3 + 0.2 * 0.7 + 0.7 * 0.9 \end{bmatrix}^T + [1.0 \quad 1.0 \quad 1.0]$$

$$=[1.35 \quad 1.27 \quad 1.8]$$

**Why Bias?**

$$\mathbf{i} * W_{ij} + \mathbf{b}_j = \mathbf{j}^{in}$$



(illustration for 1-dimensional input  
and output)

# Layer 1 – ReLU (Rectified Linear Unit) Operation

$$\text{relu}(x) = \max(0, x)$$

$$[\text{relu}(j_1^{in}) \quad \text{relu}(j_2^{in}) \quad \text{relu}(j_3^{in})] = [j_1^{out} \quad j_2^{out} \quad j_3^{out}]$$

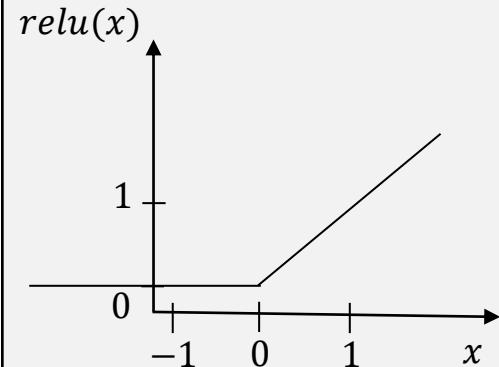
$$[\max(0, j_1^{in}) \quad \max(0, j_2^{in}) \quad \max(0, j_3^{in})] = [j_1^{out} \quad j_2^{out} \quad j_3^{out}]$$

## Example:

$$[\text{relu}(1.35) \quad \text{relu}(1.27) \quad \text{relu}(1.8)]$$

$$= [\max(0, 1.35) \quad \max(0, 1.27) \quad \max(0, 1.8)] = [1.35 \quad 1.27 \quad 1.8]$$

### ReLU Function



# Why Do I Need The Activation Function?

$$\mathbf{i} * W_{ij} + \mathbf{b}_j = \mathbf{j} = f(\mathbf{i})$$

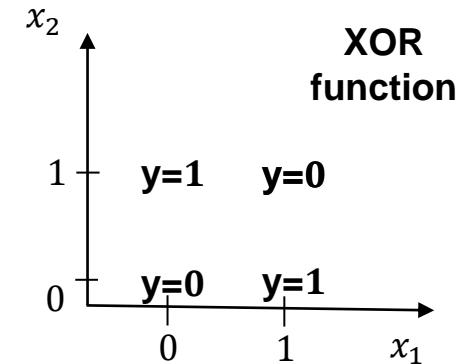
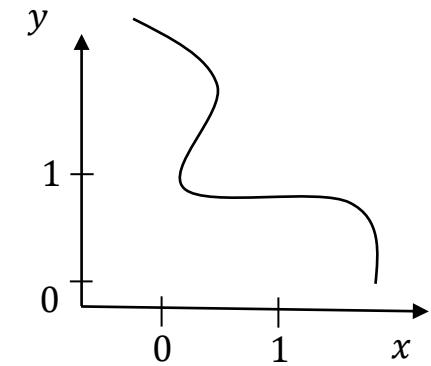
$$\mathbf{j} * W_{jk} + \mathbf{b}_k = \mathbf{k} = f'(\mathbf{j})$$

$$\mathbf{k} * W_{kl} + \mathbf{b}_l = \mathbf{l} = f''(\mathbf{k})$$

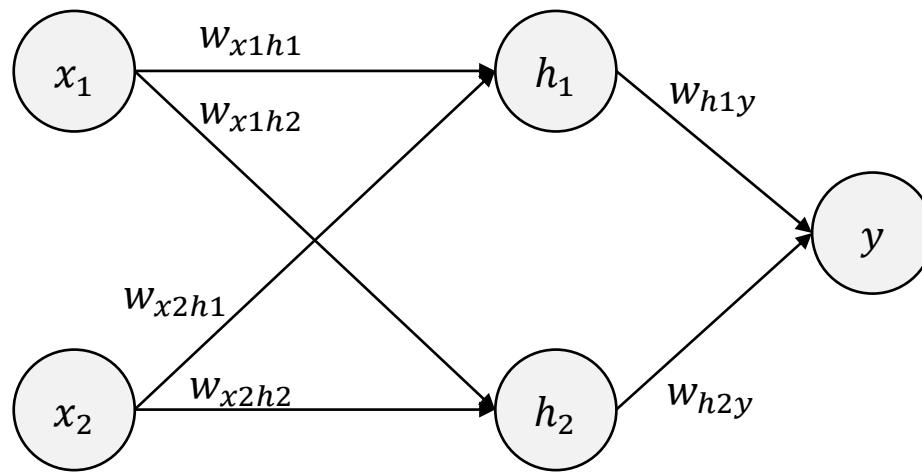
$$\mathbf{l} = f''(f'(f(\mathbf{i})))$$

$$\begin{aligned}\mathbf{l} &= ((\mathbf{i} * W_{ij} + \mathbf{b}_j) * W_{jk} + \mathbf{b}_k) * W_{kl} + \mathbf{b}_l \\ &= (\mathbf{i} * W_{ij} * W_{jk} + \mathbf{b}_j * W_{jk} + \mathbf{b}_k) * W_{kl} + \mathbf{b}_l \\ &= \underbrace{\mathbf{i} * W_{ij} * W_{jk} * W_{kl}}_{slope} + \underbrace{\mathbf{b}_j * W_{jk} * W_{kl} + \mathbf{b}_k * W_{kl} + \mathbf{b}_l}_{intercept}\end{aligned}$$

Without non-linear activation functions, such as ReLU,  $\mathbf{l} = f''(f'(f(\mathbf{i})))$  is just a linear function and not powerful enough to represent any non-linear functions.



# XOR With A 1-Layer Feed-Foward Neural Network Without Activation Function



$$y = x_1 * w_{x1h1} * w_{h1y} + x_2 * w_{x2h1} * w_{h1y} + x_1 * w_{x1h2} * w_{h2y} + x_2 * w_{x2h2} * w_{h2y} + \mathbf{b}_h * W_{hy} + \mathbf{b}_y$$

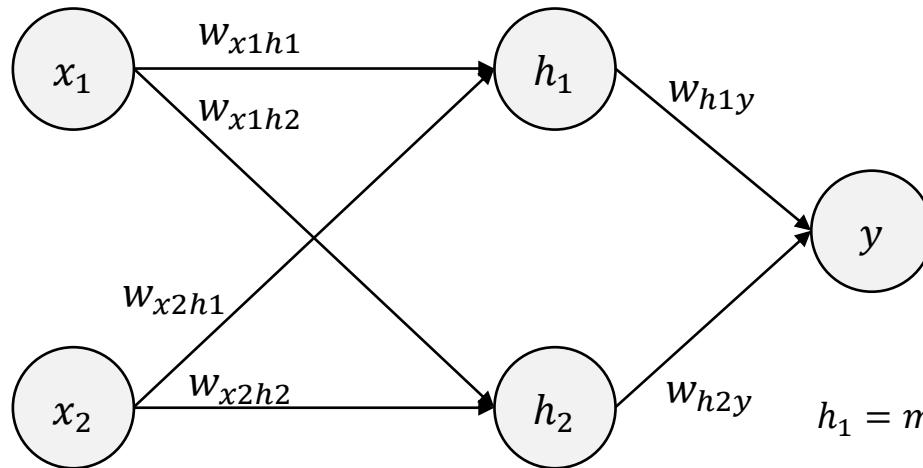
$$\text{for } x = \begin{bmatrix} 0 \\ 0 \end{bmatrix}, 0 = \mathbf{b}_h * W_{hy} + \mathbf{b}_y$$

$$\text{for } x = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, 1 = w_{x1h1} * w_{h1y} + w_{x1h2} * w_{h2y}$$

$$\text{for } x = \begin{bmatrix} 0 \\ 1 \end{bmatrix}, 1 = w_{x2h1} * w_{h1y} + w_{x2h2} * w_{h2y}$$

$$\text{for } x = \begin{bmatrix} 1 \\ 1 \end{bmatrix}, 0 \neq 1 + 1$$

# XOR With A 1-Layer Feed-Foward Neural Network With Activation Function



$$h_1 = \max(0, w_{x1h1} * x_1 + w_{x2h1} * x_2 + b_{h1}) = \max(0, x_1 + x_2)$$

$$h_2 = \max(0, w_{x1h2} * x_1 + w_{x2h2} * x_2 + b_{h2}) = \max(0, x_1 + x_2 - 1)$$

**Assume:**

$$y = \max(0, w_{h1y} * h_1 + w_{h2y} * h_2 + b_y) = \max(0, h_1 - 2 * h_2 + b_y)$$

$$= \max(0, \max(0, x_1 + x_2) - 2 * \max(0, x_1 + x_2 - 1))$$

$$\text{for } x = \begin{bmatrix} 0 \\ 0 \end{bmatrix}, 0 = \max(0, \max(0,0) - 2 * \max(0, -1))$$

$$\text{for } x = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, 1 = \max(0, \max(0,1) - 2 * \max(0, 0))$$

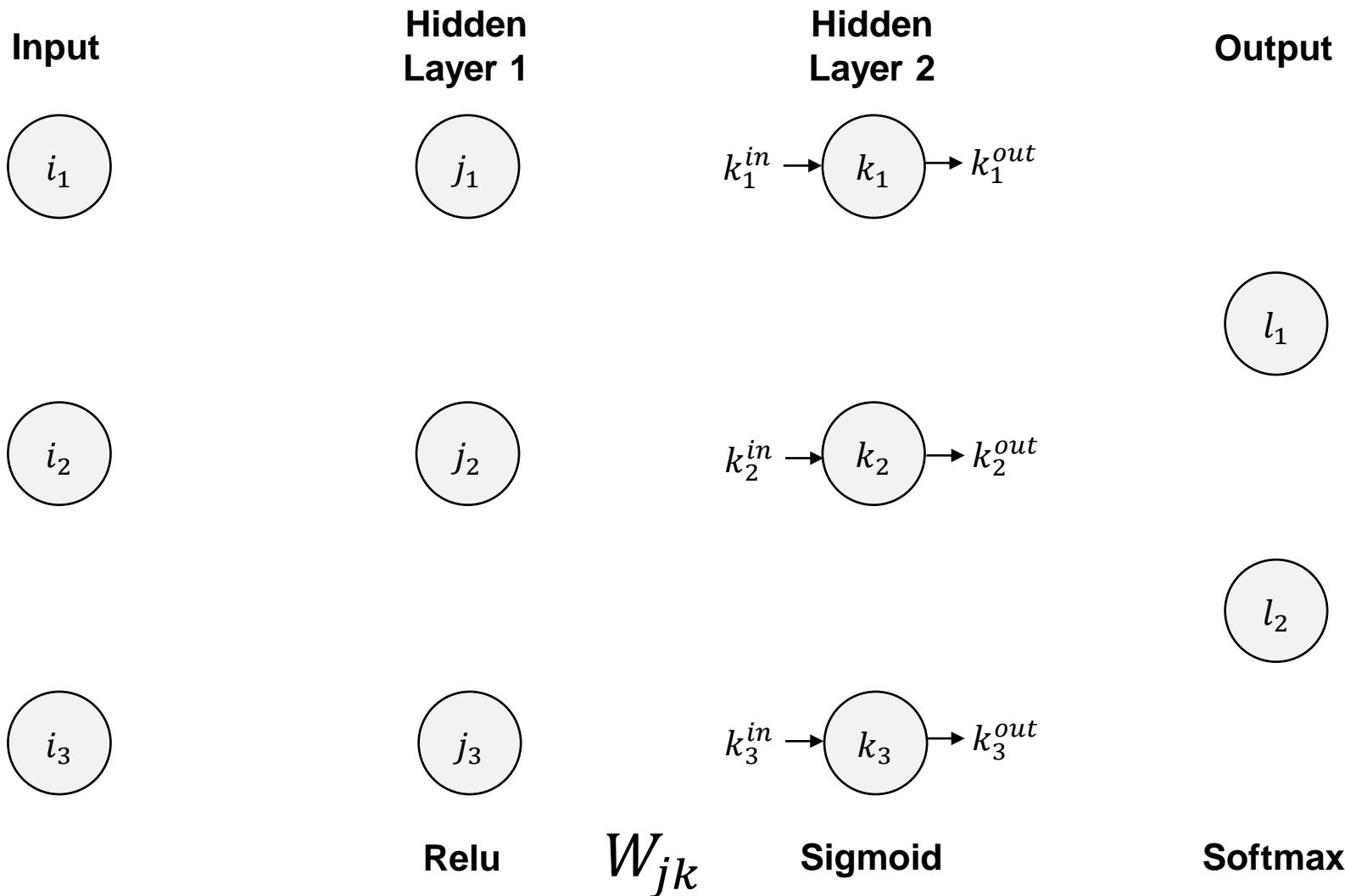
$$\text{for } x = \begin{bmatrix} 0 \\ 1 \end{bmatrix}, 1 = \max(0, \max(0,1) - 2 * \max(0, 0))$$

$$\text{for } x = \begin{bmatrix} 1 \\ 1 \end{bmatrix}, 0 = \max(0, \max(0,2) - 2 * \max(0, 1))$$

$$W_{xh} = \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}, \mathbf{w}_{hy} = \begin{bmatrix} 1 \\ -2 \end{bmatrix}$$

$$\mathbf{b}_h = \begin{bmatrix} 0 \\ -1 \end{bmatrix}, b_y = 0$$

# Layer 2



# Layer 2 – Matrix Operation

$$\mathbf{j}^{out} * W_{jk} + \mathbf{b}_k = \mathbf{k}^{in}$$

$$[j_1^{out} \quad j_2^{out} \quad j_3^{out}] * \begin{bmatrix} w_{j1k1} & w_{j1k2} & w_{j1k3} \\ w_{j2k1} & w_{j2k2} & w_{j2k3} \\ w_{j3k1} & w_{j3k2} & w_{j3k3} \end{bmatrix} + [b_{k1} \quad b_{k2} \quad b_{k3}] = [k_1^{in} \quad k_2^{in} \quad k_3^{in}]$$

$$\begin{bmatrix} j_1^{out} * w_{j1k1} + j_2^{out} * w_{j2k1} + j_3^{out} * w_{j3k1} \\ j_1^{out} * w_{j1k2} + j_2^{out} * w_{j2k2} + j_3^{out} * w_{j3k2} \\ j_1^{out} * w_{j1k3} + j_2^{out} * w_{j2k3} + j_3^{out} * w_{j3k3} \end{bmatrix}^T + [b_{k1} \quad b_{k2} \quad b_{k3}] = [k_1^{in} \quad k_2^{in} \quad k_3^{in}]$$

## Example:

$$[1.35 \quad 1.27 \quad 1.8] * \begin{bmatrix} 0.2 & 0.3 & 0.5 \\ 0.3 & 0.5 & 0.7 \\ 0.6 & 0.4 & 0.8 \end{bmatrix} + [1.0 \quad 1.0 \quad 1.0]$$

$$= \begin{bmatrix} 1.35 * 0.2 + 1.27 * 0.3 + 1.8 * 0.6 \\ 1.35 * 0.3 + 1.27 * 0.5 + 1.8 * 0.4 \\ 1.35 * 0.5 + 1.27 * 0.7 + 1.8 * 0.8 \end{bmatrix}^T + [1.0 \quad 1.0 \quad 1.0]$$

$$=[2.73 \quad 2.76 \quad 4]$$

# Layer 2 – Sigmoid Operation

$$\text{sigmoid}(x) = 1/(1 + e^{-x})$$

$$[\text{sigmoid}(k_1^{in}) \quad \text{sigmoid}(k_2^{in}) \quad \text{sigmoid}(k_3^{in})] = [k_1^{out} \quad k_2^{out} \quad k_3^{out}]$$

$$\left[ 1/\left(1 + e^{-k_1^{in}}\right) \quad 1/\left(1 + e^{-k_2^{in}}\right) \quad 1/\left(1 + e^{-k_3^{in}}\right) \right] = [k_1^{out} \quad k_2^{out} \quad k_3^{out}]$$

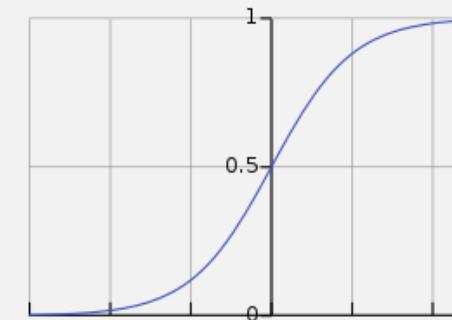
**Example:**

$$[\text{sigmoid}(2.73) \quad \text{sigmoid}(2.76) \quad \text{sigmoid}(4)]$$

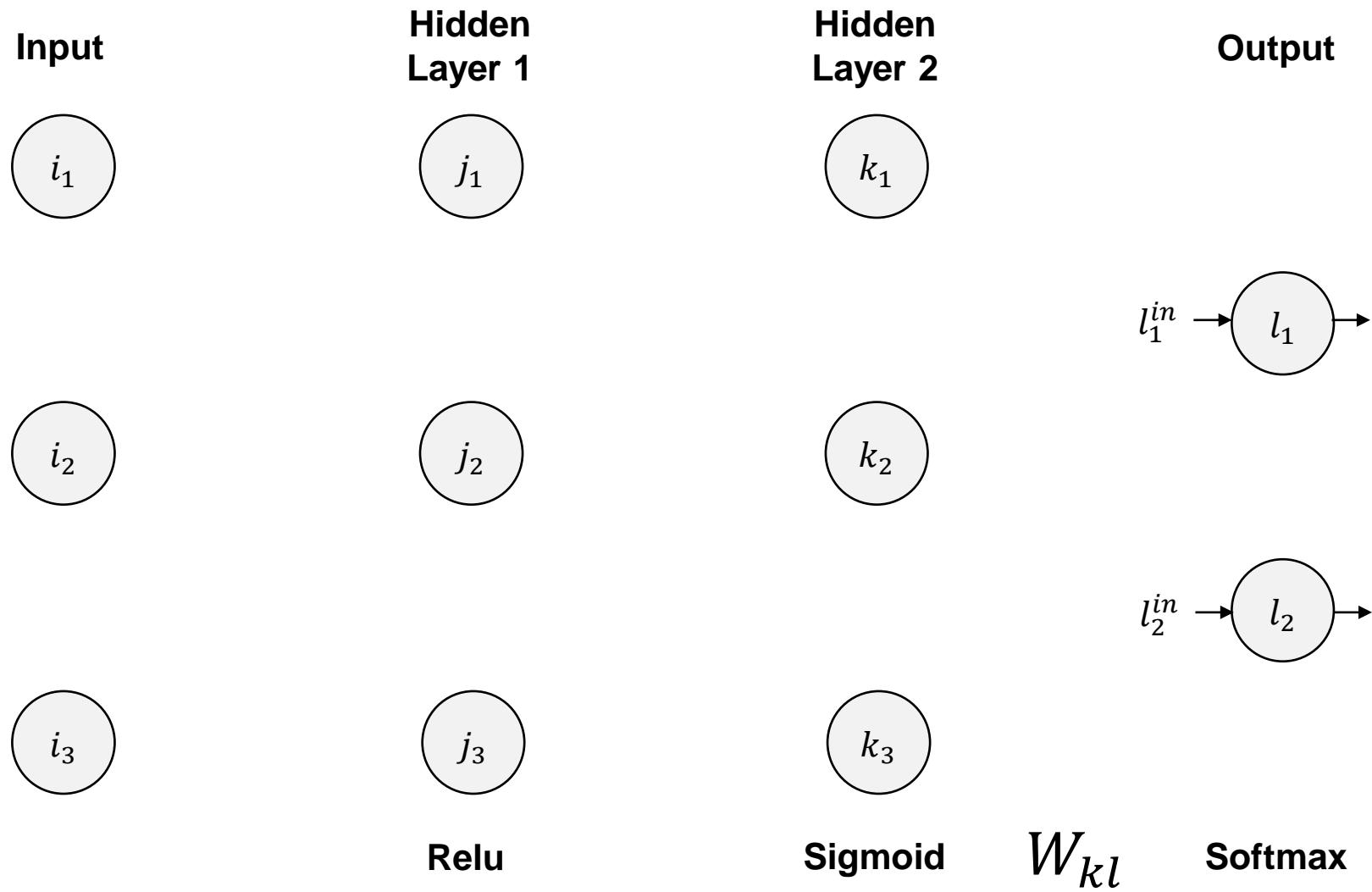
$$= [1/(1 + e^{-2.73}) \quad 1/(1 + e^{-2.76}) \quad 1/(1 + e^{-4})] = [0.938 \quad 0.94 \quad 0.982]$$

**Sigmoid Function**

$$\text{sigmoid}(x)$$



# Output Layer



# Layer 3 – Matrix Operation

$$k^{out} * W_{kl} + b_l = l^{in}$$

$$\begin{bmatrix} k_1^{out} & k_2^{out} & k_3^{out} \end{bmatrix} * \begin{bmatrix} w_{k1l1} & w_{k1l2} \\ w_{k2l1} & w_{k2l2} \\ w_{k3l1} & w_{k3l2} \end{bmatrix} + [b_{l1} \quad b_{l2}] = [l_1^{in} \quad l_2^{in}]$$

$$\begin{bmatrix} k_1^{out} * w_{k1l1} + k_2^{out} * w_{k2l1} + k_3^{out} * w_{k3l1} \\ k_1^{out} * w_{k1l2} + k_2^{out} * w_{k2l2} + k_3^{out} * w_{k3l2} \end{bmatrix}^T + [b_{l1} \quad b_{l2}] = [l_1^{in} \quad l_2^{in}]$$

**Example:**

$$\begin{bmatrix} 0.938 & 0.94 & 0.982 \end{bmatrix} * \begin{bmatrix} 0.1 & 0.4 \\ 0.3 & 0.7 \\ 0.5 & 0.2 \end{bmatrix} + [1.0 \quad 1.0]$$

$$= \begin{bmatrix} 0.938 * 0.1 + 0.94 * 0.3 + 0.982 * 0.5 \\ 0.938 * 0.4 + 0.94 * 0.7 + 0.982 * 0.2 \end{bmatrix}^T + [1.0 \quad 1.0]$$

$$=[0.8668 \quad 1.2296]$$

# Layer 3 – Softmax Operation

$$\text{softmax}(\mathbf{x})_i = e^{x_i} / \sum_{j=1}^K e^{x_j} \text{ for } i = 1, \dots, K \text{ and } \mathbf{x} = (x_1, \dots, x_K)$$

$$[\text{softmax}(\mathbf{l}^{in})_1 \quad \text{softmax}(\mathbf{l}^{in})_2] = [l_1^{out} \quad l_2^{out}]$$

$$\left[ e^{l_1^{in}} / \sum_{j=1}^2 e^{l_j^{in}} \quad e^{l_2^{in}} / \sum_{j=1}^2 e^{l_j^{in}} \right] = [l_1^{out} \quad l_2^{out}]$$

**Example:**

$$\begin{aligned} & [\text{softmax}([0.8668 \quad 1.2296])_1 \quad \text{softmax}([0.8668 \quad 1.2296])_2] \\ &= [e^{0.8668} / e^{0.8668} + e^{1.2296} \quad e^{1.2296} / e^{0.8668} + e^{1.2296}] \\ &= [0.41 \quad 0.59] \end{aligned}$$

# Calculating Error

$$\text{Actual Output} = \hat{\mathbf{l}} = [1.0 \quad 0.0] \quad \text{Predicted Output} = \mathbf{l}^{out} = [0.41 \quad 0.59]$$

$$\text{Mean Squared Error (MSE)} = \frac{1}{n} \sum_{i=1}^n (l_i^{out} - \hat{l}_i)^2$$

$$\text{Mean Squared Error (MSE)} = \frac{1}{2} \left( (l_1^{out} - \hat{l}_1)^2 + (l_2^{out} - \hat{l}_2)^2 \right)$$

## Example:

$$\text{Mean Squared Error (MSE)} = \frac{1}{2} \left( (0.41 - 1.0)^2 + (0.59 - 0.0)^2 \right) = 0.3481$$

# Some Recomendations

Problem Type	Output Type	Final Activation Function	Loss Function
Regression	Numerical value	Linear	Mean Squared Error (MSE)
Classification	Binary outcome	Sigmoid	Binary Cross Entropy
Classification	Single label, multiple classes	Softmax	Cross Entropy
Classification	Multiple labels, multiple classes	Sigmoid	Binary Cross Entropy

# Backward Propagation

- Adjust each weight in order to decrease error
- How much changes error if  $l_1^{out}$  changes?

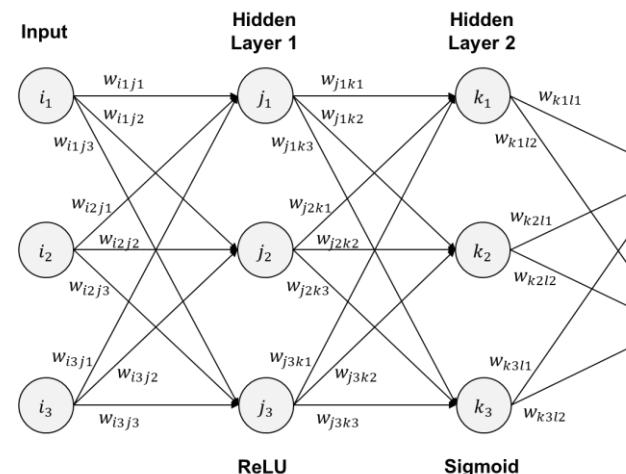
$$\frac{\partial MSE}{\partial l_1^{out}} = \frac{\partial^{\frac{1}{2}}((l_1^{out} - \hat{l}_1)^2 + (l_2^{out} - \hat{l}_2)^2)}{\partial l_1^{out}}$$

$$= \frac{\partial^{\frac{1}{2}}((l_1^{out})^2 - 2l_1^{out}\hat{l}_1 + \hat{l}_1^2) + ((l_2^{out})^2 - 2l_2^{out}\hat{l}_2 + \hat{l}_2^2)}{\partial l_1^{out}}$$

$$= \frac{\partial^{\frac{1}{2}}l_1^{out}^2}{\partial l_1^{out}} - \frac{\partial^{\frac{1}{2}} * 2l_1^{out}\hat{l}_1}{\partial l_1^{out}} = l_1^{out} - \hat{l}_1$$

## Example:

$$\begin{bmatrix} \frac{\partial MSE}{\partial l_1^{out}} & \frac{\partial MSE}{\partial l_2^{out}} \end{bmatrix} = [0.41 - 1.0 \quad 0.59 - 0.0] = [-0.59 \quad 0.59]$$



# Backward Propagation

- How much changes  $l_1^{out}$  if  $l_1^{in}$  changes?

$$\frac{\partial l_1^{out}}{\partial l_1^{in}} = \frac{\partial e^{l_1^{in}} / \sum_{j=1}^2 e^{l_j^{in}}}{\partial l_1^{in}} = \frac{\partial e^{l_1^{in}} / (e^{l_1^{in}} + e^{l_2^{in}})}{\partial l_1^{in}}$$

**Quotient Rule:**  
 $\left[ \frac{u(x)}{v(x)} \right]' = \frac{u(x)' * v(x) - u(x) * v'(x)}{v(x)^2}$

$$= \frac{\frac{\partial e^{l_1^{in}}}{\partial l_1^{in}} * (e^{l_1^{in}} + e^{l_2^{in}}) - e^{l_1^{in}} * \frac{\partial e^{l_1^{in}} + e^{l_2^{in}}}{\partial l_1^{in}}}{(e^{l_1^{in}} + e^{l_2^{in}})^2}$$

$$= \frac{e^{l_1^{in}} * (e^{l_1^{in}} + e^{l_2^{in}}) - e^{l_1^{in}} * e^{l_1^{in}}}{(e^{l_1^{in}} + e^{l_2^{in}})^2} = \frac{e^{l_1^{in}} * (e^{l_1^{in}} + e^{l_2^{in}}) - e^{2 * l_1^{in}}}{(e^{l_1^{in}} + e^{l_2^{in}})^2} = \frac{e^{l_1^{in}} + e^{l_2^{in}}}{(e^{l_1^{in}} + e^{l_2^{in}})^2}$$

## Example:

$$\begin{bmatrix} \frac{\partial l_1^{out}}{\partial l_1^{in}} & \frac{\partial l_2^{out}}{\partial l_2^{in}} \end{bmatrix} = \begin{bmatrix} \frac{e^{0.8668+1.2296}}{(e^{0.8668}+e^{1.2296})^2} & \frac{e^{0.8668+1.2296}}{(e^{0.8668}+e^{1.2296})^2} \end{bmatrix} = [0.242 \quad 0.242]$$

# Backward Propagation

- How much changes  $l_1^{in}$  if  $w_{k1l1}$  changes?

$$\frac{\partial l_1^{in}}{\partial w_{k1l1}} = \frac{\partial k_1^{out} * w_{k1l1} + k_2^{out} * w_{k2l1} + k_3^{out} * w_{k3l1} + b_{l1}}{\partial w_{k1l1}} = k_1^{out}$$

## Example:

$$\begin{bmatrix} \frac{\partial l_1^{in}}{\partial w_{k1l1}} & \frac{\partial l_1^{in}}{\partial w_{k2l1}} & \frac{\partial l_1^{in}}{\partial w_{k3l1}} \end{bmatrix} = [k_1^{out} \quad k_2^{out} \quad k_3^{out}] = [0.938 \quad 0.94 \quad 0.982]$$

$$\begin{bmatrix} \frac{\partial l_2^{in}}{\partial w_{k1l2}} & \frac{\partial l_2^{in}}{\partial w_{k2l2}} & \frac{\partial l_2^{in}}{\partial w_{k3l2}} \end{bmatrix} = [k_1^{out} \quad k_2^{out} \quad k_3^{out}] = [0.938 \quad 0.94 \quad 0.982]$$

# Backward Propagation

- Now we can finally calculate how to adjust the weights  $W_{kl}$
- How much changes the error if  $w_{k1l1}$  changes?

$$\frac{\partial MSE}{\partial w_{k1l1}} = \frac{\partial MSE}{\partial l_1^{out}} * \frac{\partial l_1^{out}}{\partial l_1^{in}} * \frac{\partial l_1^{in}}{\partial w_{k1l1}}$$

**Chain Rule:**

$$\frac{\partial z}{\partial x} = \frac{\partial z}{\partial y} * \frac{\partial y}{\partial x}$$

$$= (l_1^{out} - \hat{l}_1) * \frac{e^{l_1^{in}} + e^{l_2^{in}}}{(e^{l_1^{in}} + e^{l_2^{in}})^2} * k_1^{out}$$

## Example:

$$\frac{\partial MSE}{\partial w_{k1l1}} = -0.59 * 0.242 * 0.938 = -0.134$$

# Updating Weights

- Weights are updated by subtracting  $\frac{\partial MSE}{\partial w_{k1l1}}$  from  $w_{k1l1}$  under consideration of the learning rate  $\alpha$  (in the example we choose  $\alpha = 0.01$ ).

$$\tilde{w}_{k1l1} = w_{k1l1} - \alpha * \frac{\partial MSE}{\partial w_{k1l1}}$$

## Example:

$$\tilde{w}_{k1l1} = 0.1 - 0.01 * (-0.134) = 0.10134$$

# Updating Bias

- How much changes  $l_1^{in}$  if  $b_{l1}$  changes?

$$\frac{\partial l_1^{in}}{\partial b_{l1}} = \frac{\partial k_1^{out} * w_{k1l1} + k_2^{out} * w_{k2l1} + k_3^{out} * w_{k3l1} + b_{l1}}{\partial b_{l1}} = 1$$

$$\frac{\partial MSE}{\partial b_{l1}} = \frac{\partial MSE}{\partial l_1^{out}} * \frac{\partial l_1^{out}}{\partial l_1^{in}} * \frac{\partial l_1^{in}}{\partial b_{l1}}$$

$$= (l_1^{out} - \hat{l}_1) * \frac{e^{l_1^{in}} + e^{l_2^{in}}}{(e^{l_1^{in}} + e^{l_2^{in}})^2} * 1$$

- Update bias the same way as the weights

$$\tilde{b}_{l1} = b_{l1} - \alpha * \frac{\partial MSE}{\partial b_{l1}}$$

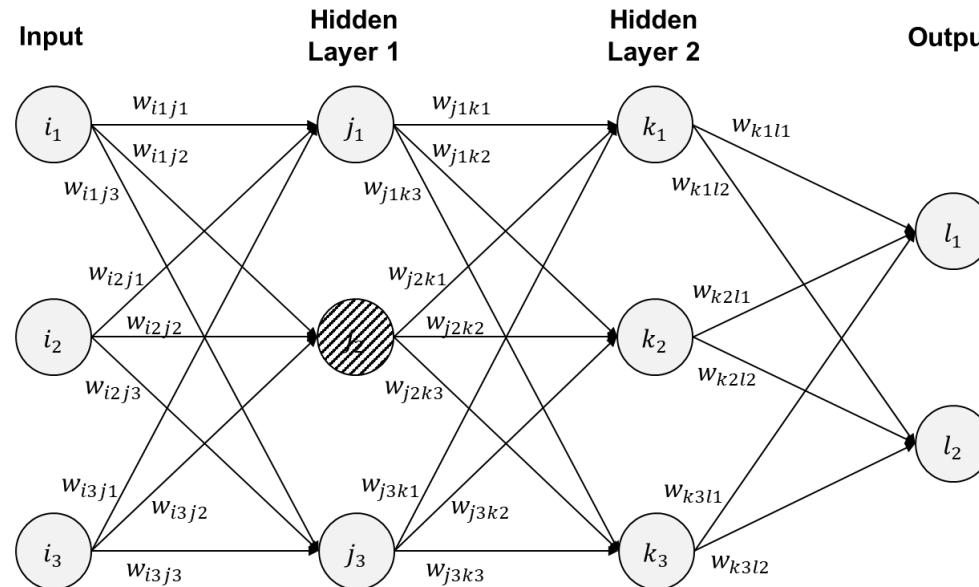
## Example:

$$\frac{\partial MSE}{\partial b_{l1}} = -0.59 * 0.242 * 1 = -0.1427$$

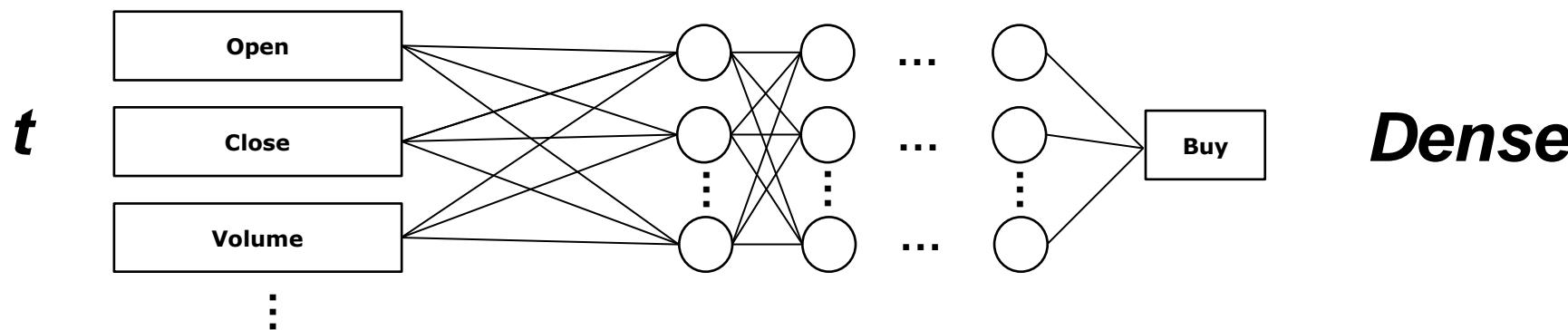
$$\tilde{b}_{l1} = 1.0 - 0.01 * (-0.1427) = 1.0014$$

# Dropout

- Problem of Overfitting: Too many different models can be fit to the data
- Solution: Fit all possible models to the same data and average predictions from each model (= an approximation is ensemble learning)
- Problem: Requires multiple models to be fit
- Dropout: Randomly drop nodes during training → Approximates training large number of neural networks with different architecture parallel

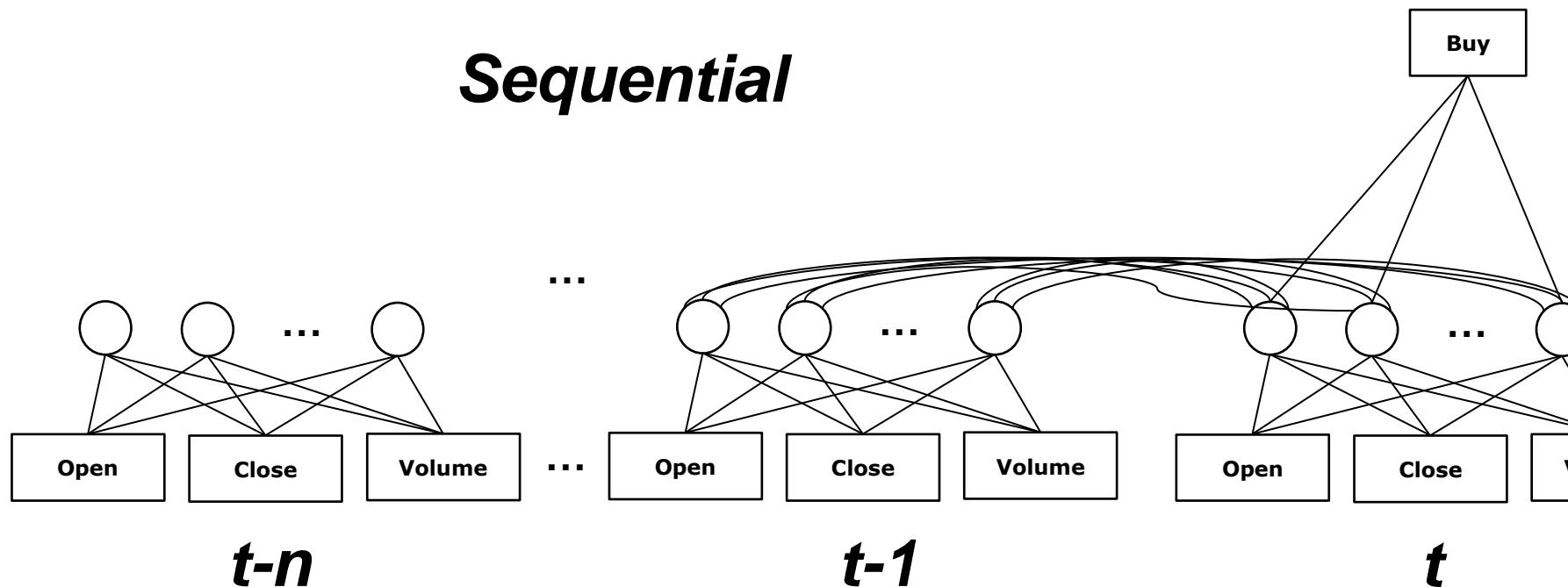


# Dense Vs. Sequential



**Dense**

**Sequential**



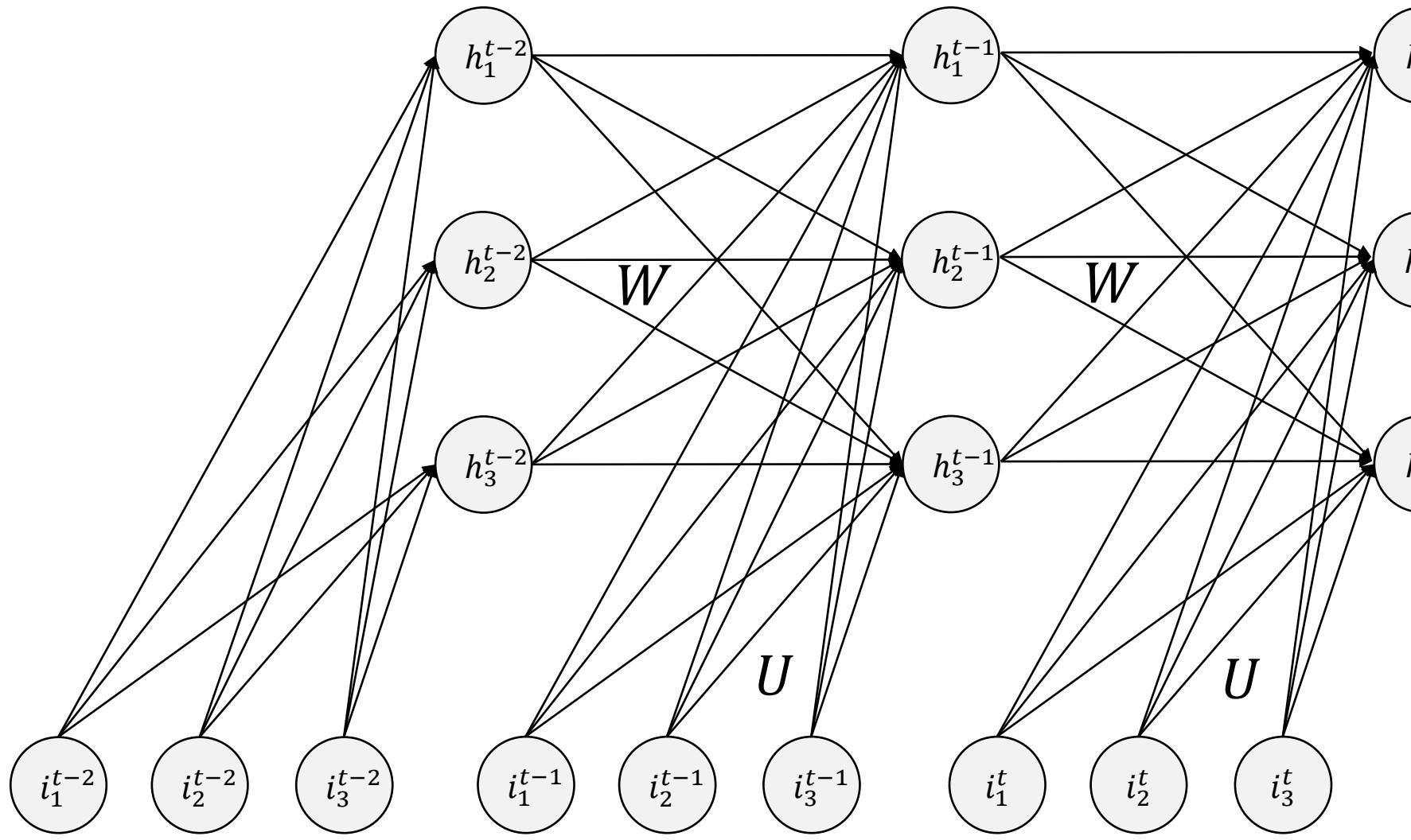
***t-n***

***t-1***

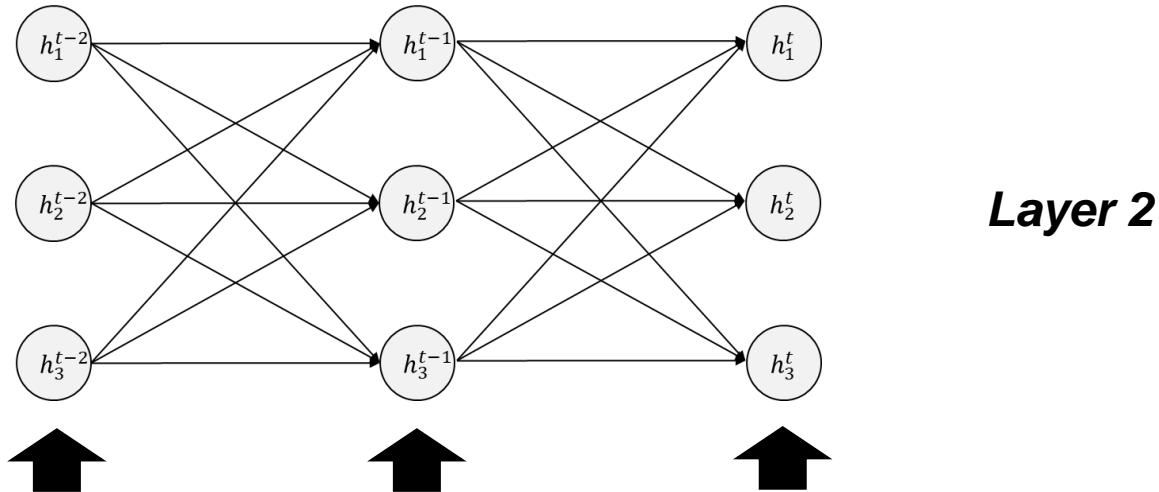
***t***

# Recurrent Neural Networks (RNN)

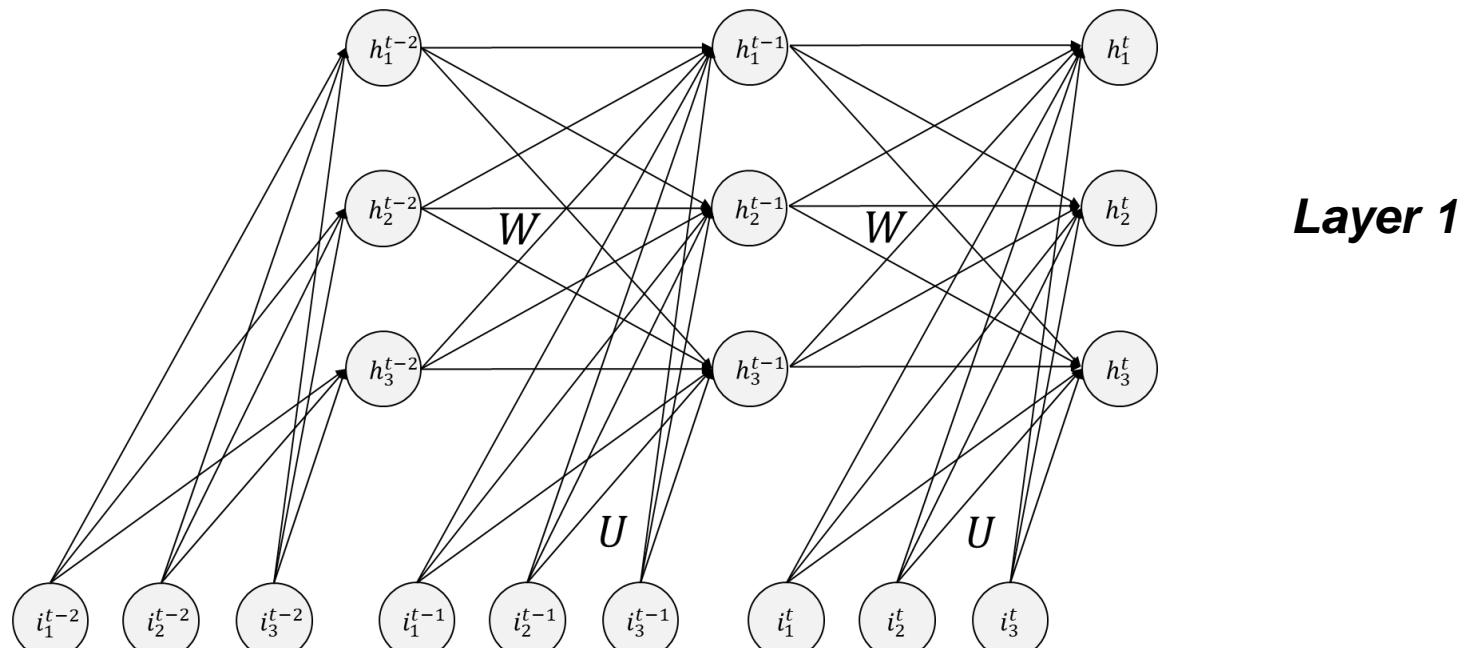
Update step from  $t - 1$  to  $t$ :  $\mathbf{i}^t * U + \mathbf{h}^{t-1} * W + \mathbf{b} =$



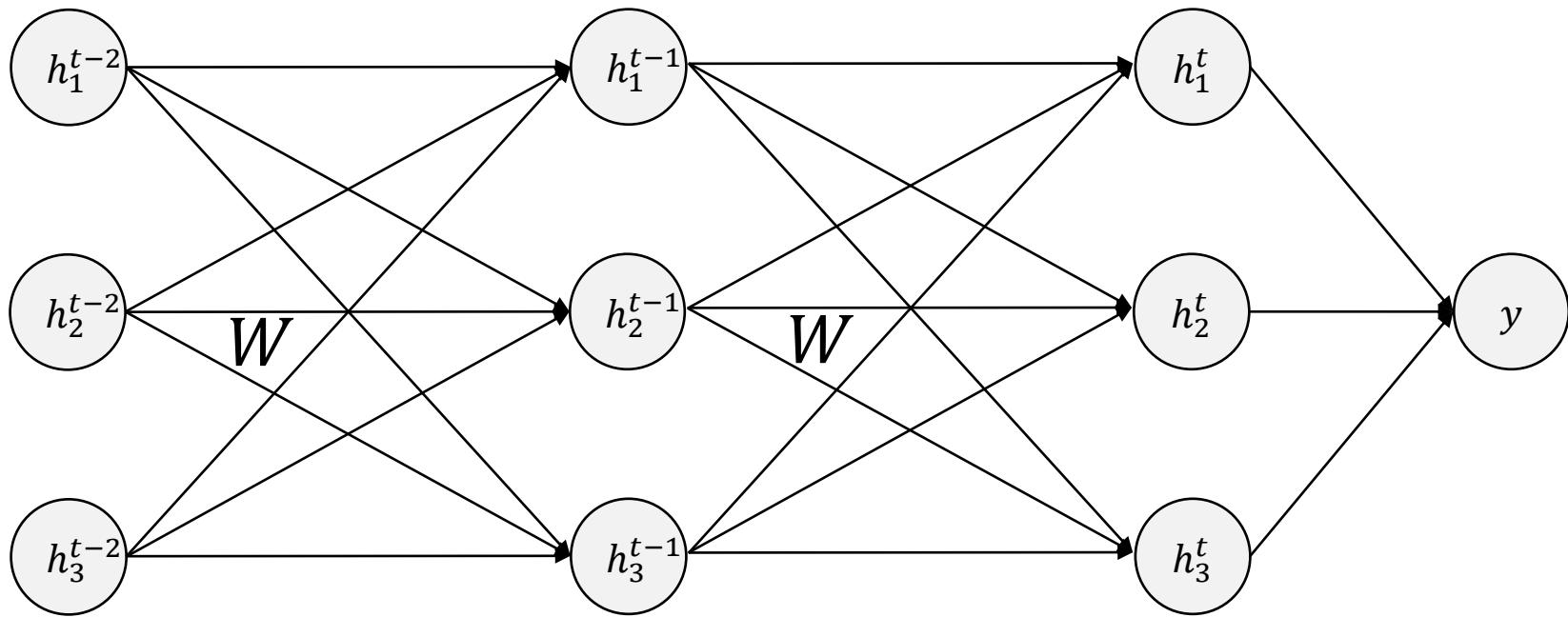
# Multiple Layers of RNNs



Output sequence of layer  
1 is input of layer 2



# Backward Propagation Through Time

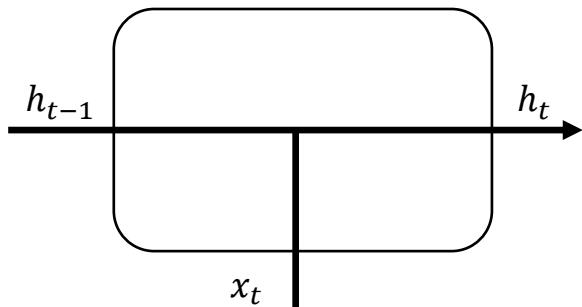


$$\frac{\partial MSE}{\partial W} = \sum_{k=1}^t \frac{\partial MSE}{\partial y} * \frac{\partial y}{\partial h^t} * \frac{\partial h^t}{\partial h^k} * \frac{\partial h^k}{\partial W}$$

$$\frac{\partial MSE}{\partial W} = \sum_{k=1}^t \frac{\partial MSE}{\partial y} * \frac{\partial y}{\partial h^t} * \left( \prod_{j=k+1}^t \frac{\partial h^j}{\partial h^{j-1}} \right) * \frac{\partial h^k}{\partial W}$$

# RNN vs. Long Short Term Memory (LSTM)

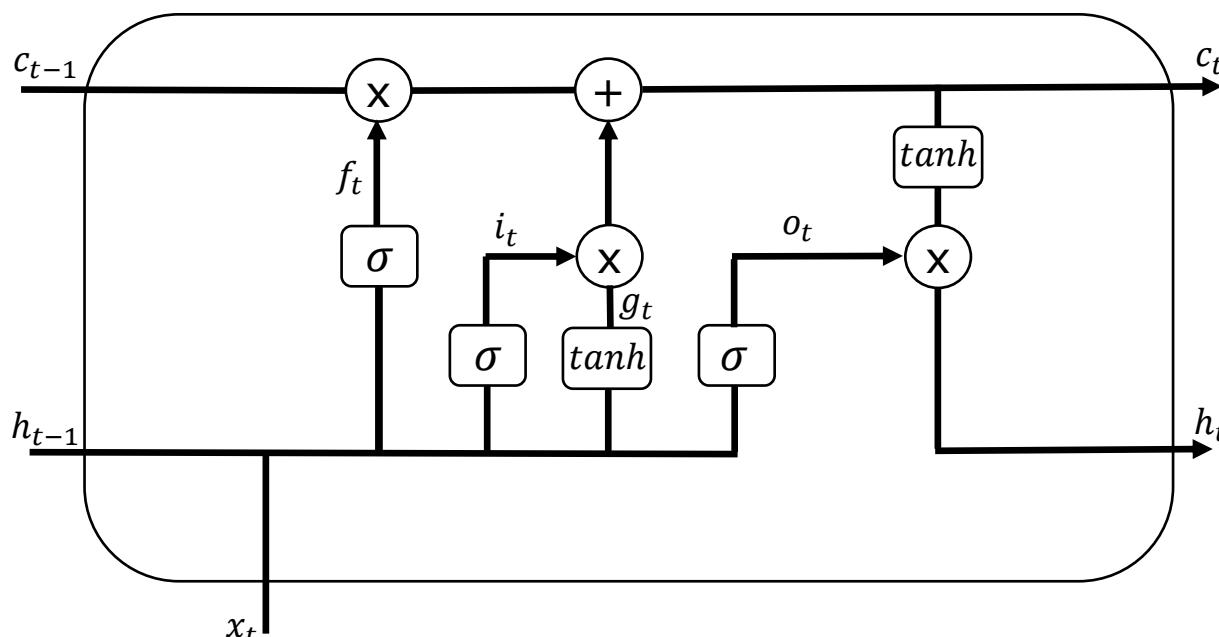
**RNN cell**



Update step from  $t - 1$  to  $t$

$$h_t = W h_{t-1} + U x_t + b$$

**LSTM cell**



$$f_t = \sigma(U_f x_t + V_f h_{t-1} + b_f)$$

$$i_t = \sigma(U_i x_t + V_i h_{t-1} + b_i)$$

$$o_t = \sigma(U_o x_t + V_o h_{t-1} + b_o)$$

$$g_t = \tanh(U_g x_t + V_g h_{t-1})$$

$$c_t = f_t \circ c_{t-1} + i_t \circ g_t$$

$$h^t = o_t \circ \tanh(c_t)$$

○ denotes element-wise multiplication

# Convolution

*2-Dimesional Input: I*

13	15	7	2	16	25
12	18	9	6	12	2
15	1	10	0	34	2
18	16	3	23	13	15
10	17	1	9	12	16
12	1	12	6	2	23

*Filter: W*

1	0	-1
1	0	-1
1	0	-1

*Output: J*


# Convolution

2-Dimesional Input:  $I$

13	15	7	2	16	25
12	18	9	6	12	2
15	1	10	0	34	2
18	16	3	23	13	15
10	17	1	9	12	16
12	1	12	6	2	23

Filter:  $W$

1	0	-1
1	0	-1
1	0	-1

Output:  $J$

14		

Applying the filter:

$$\sum_{i,j} \left( \begin{bmatrix} 13 & 15 & 7 \\ 12 & 18 & 9 \\ 15 & 1 & 10 \end{bmatrix} \circ \begin{bmatrix} 1 & 0 & -1 \\ 1 & 0 & -1 \\ 1 & 0 & -1 \end{bmatrix} \right) =$$

$$\begin{aligned}
 & 13 * 1 + 15 * 0 + 7 * (-1) \\
 & + 12 * 1 + 18 * 0 + 9 * (-1) \\
 & + 15 * 1 + 1 * 0 + 10 * (-1)
 \end{aligned}$$

$$= 14$$

# Convolution

2-Dimesional Input:  $I$

13	15	7	2	16	25
12	18	9	6	12	2
15	1	10	0	34	2
18	16	3	23	13	15
10	17	1	9	12	16
12	1	12	6	2	23

Filter:  $W$

1	0	-1
1	0	-1
1	0	-1

Output:  $J$

14	26		

Applying the filter:

$$\sum_{i,j} \left( \begin{bmatrix} 15 & 7 & 2 \\ 18 & 9 & 6 \\ 1 & 10 & 0 \end{bmatrix} \circ \begin{bmatrix} 1 & 0 & -1 \\ 1 & 0 & -1 \\ 1 & 0 & -1 \end{bmatrix} \right) =$$

$$\begin{aligned}
 & 15 * 1 + 7 * 0 + 2 * (-1) \\
 & + 18 * 1 + 9 * 0 + 6 * (-1) \\
 & + 1 * 1 + 10 * 0 + 0 * (-1)
 \end{aligned}$$

$$= 26$$

# Convolution

2-Dimesional Input:  $I$

13	15	7	2	16	25
12	18	9	6	12	2
15	1	10	0	34	2
18	16	3	23	13	15
10	17	1	9	12	16
12	1	12	6	2	23

Convolution



Filter:  $W$

1	0	-1
1	0	-1
1	0	-1

Output:  $J$

=

14	26	-36
23	6	-37
29	2	-45
24	-4	-11

In pytorch:

`m = nn.functional.conv2d(kernel_size=(3,3))`

`output = m(input)`

# Padding

*2-Dimesional Input*

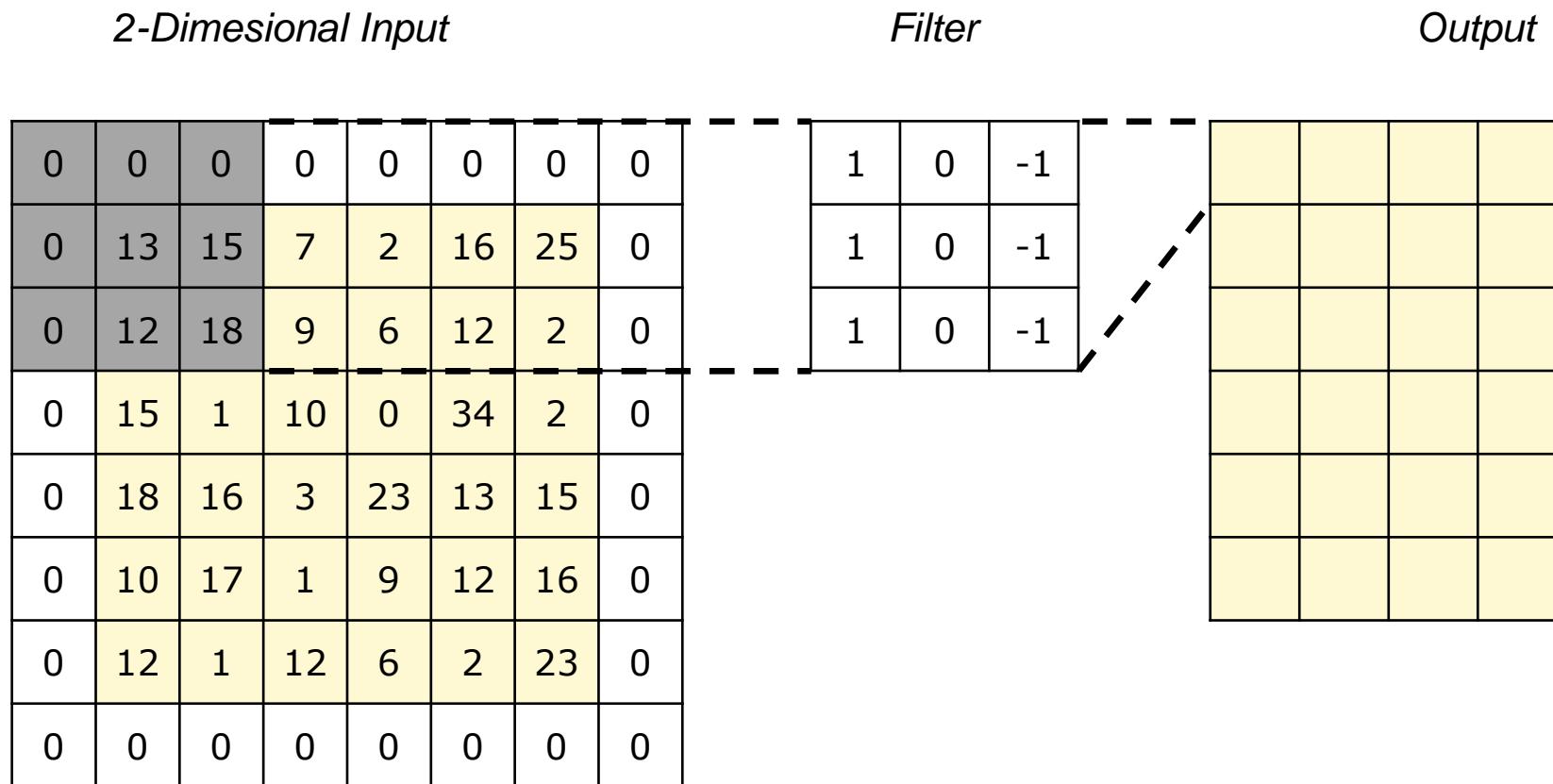
0	0	0	0	0	0	0	0
0	13	15	7	2	16	25	0
0	12	18	9	6	12	2	0
0	15	1	10	0	34	2	0
0	18	16	3	23	13	15	0
0	10	17	1	9	12	16	0
0	12	1	12	6	2	23	0
0	0	0	0	0	0	0	0

*Filter*

1	0	-1
1	0	-1
1	0	-1

*Output*


# Padding



In pytorch:

`m = nn.functional.conv2d(kernel_size=(3,3), padding=1)`

`output = m(input)`

# Stride (with Stride=2)

2-Dimesional Input:  $I$

13	15	7	2	16	25
12	18	9	6	12	2
15	1	10	0	34	2
18	16	3	23	13	15
10	17	1	9	12	16
12	1	12	6	2	23

Filter:  $W$

1	0	-1
1	0	-1
1	0	-1

Output:  $J$

14		

# Stride (with Stride=2)

2-Dimesional Input:  $I$

13	15	7	2	16	25
12	18	9	6	12	2
15	1	10	0	34	2
18	16	3	23	13	15
10	17	1	9	12	16
12	1	12	6	2	23

Filter:  $W$

1	0	-1
1	0	-1
1	0	-1

Output:  $J$

14	-36	

In pytorch:

`m = nn.functional.conv2d(kernel_size=(3,3), padding=1, stride=2, output=m(input))`

# Pooling (3x3 Maximum Pooling)

2-Dimesional Input:  $I$

13	15	7	2	16	25
12	18	9	6	12	2
15	1	10	0	34	2
18	16	3	23	13	15
10	17	1	9	12	16
12	1	12	6	2	23

Output:  $J$

18	

# Pooling (3x3 Maximum Pooling)

2-Dimesional Input:  $I$

13	15	7	2	16	25
12	18	9	6	12	2
15	1	10	0	34	2
18	16	3	23	13	15
10	17	1	9	12	16
12	1	12	6	2	23

Output:  $J$

18	34

# Pooling (3x3 Maximum Pooling)

2-Dimesional Input:  $I$

13	15	7	2	16	25
12	18	9	6	12	2
15	1	10	0	34	2
18	16	3	23	13	15
10	17	1	9	12	16
12	1	12	6	2	23

Output:  $J$

18	34
18	

# Pooling (3x3 Maximum Pooling)

2-Dimesional Input:  $I$

13	15	7	2	16	25
12	18	9	6	12	2
15	1	10	0	34	2
18	16	3	23	13	15
10	17	1	9	12	16
12	1	12	6	2	23

Output:  $J$

18	34
18	23

In pytorch:

$m = nn.MaxPool2d(3)$

$output = m(input)$

# Multiple Input Channels (e.g. RGB values)

0	0	0	0	0	0	0	...
0	156	155	156	158	158	158	...
0	153	154	157	159	159	159	...
0	149	151	155	158	159	159	...
0	146	146	149	153	153	158	...
0	145	143	143	148	158	158	...
...	...	...	...	...	...	...	...

Input Channel #1 (Red)

0	0	0	0	0	0	0	...
0	167	166	167	169	169	169	...
0	164	165	168	170	170	170	...
0	160	162	166	169	170	170	...
0	156	156	159	163	168	168	...
0	155	153	153	158	168	168	...
...	...	...	...	...	...	...	...

Input Channel #2 (Green)

0	0	0	0	0	0	0	...
0	163	162	163	165	165	165	...
0	160	161	164	166	166	166	...
0	156	158	162	165	166	166	...
0	155	155	158	162	167	167	...
0	154	152	152	157	167	167	...
...	...	...	...	...	...	...	...

Input Channel #3 (Blue)

-1	-1	1
0	1	-1
0	1	1

Kernel Channel #1



308

1	0	0
1	-1	-1
1	0	-1

Kernel Channel #2



-498

0	1	1
0	1	0
1	-1	1

Kernel Channel #3



164

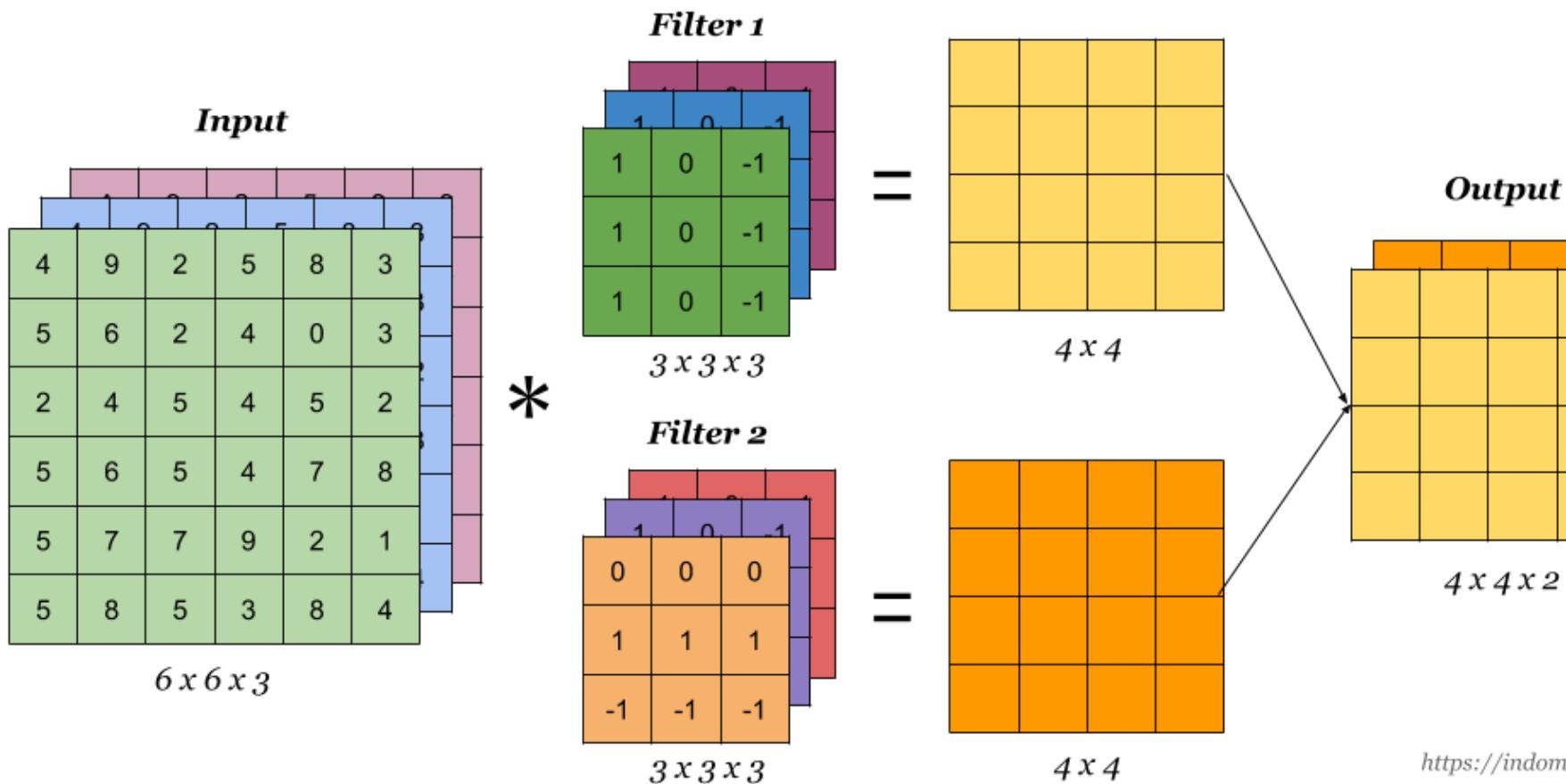
-25			
...	...	...	...

Bias = 1

In pytorch:

```
m = nn.functional.conv2d(in_channels=3, kernel_size=3, padding=1)
output = m(input)
```

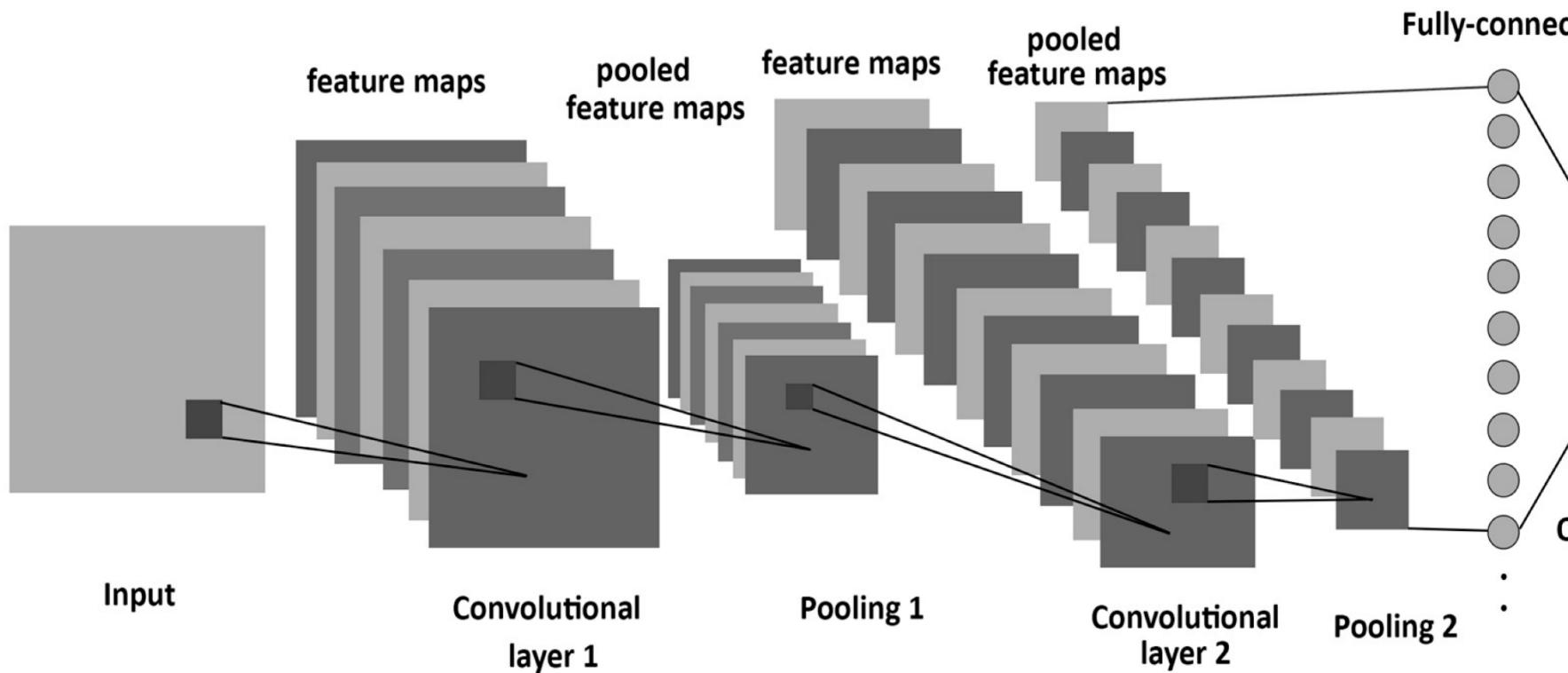
# Multiple Filters (Output Channels)



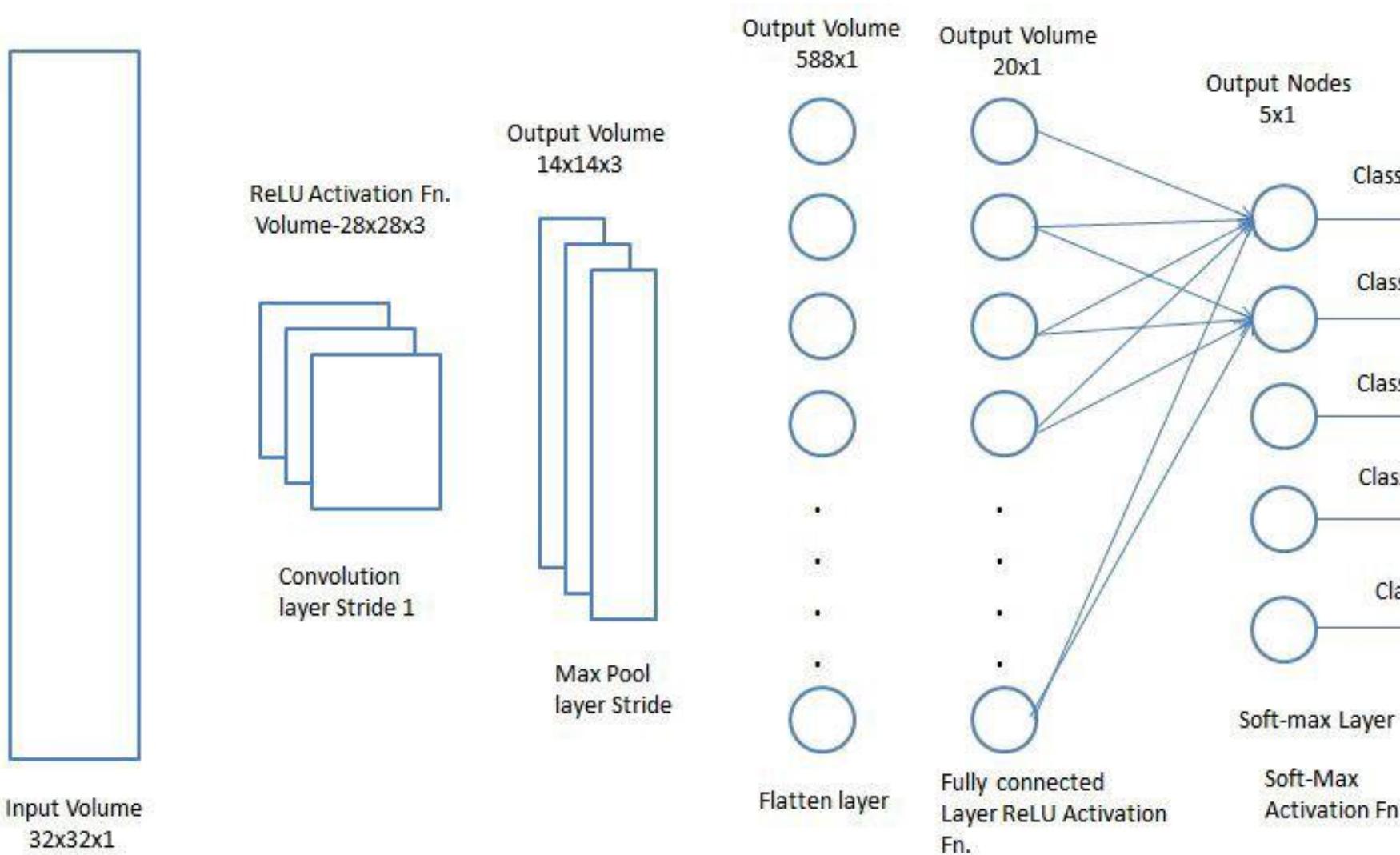
In pytorch:

`m = nn.functional.conv2d(in_channels=3, out_channels=2, kernel_size=3)`

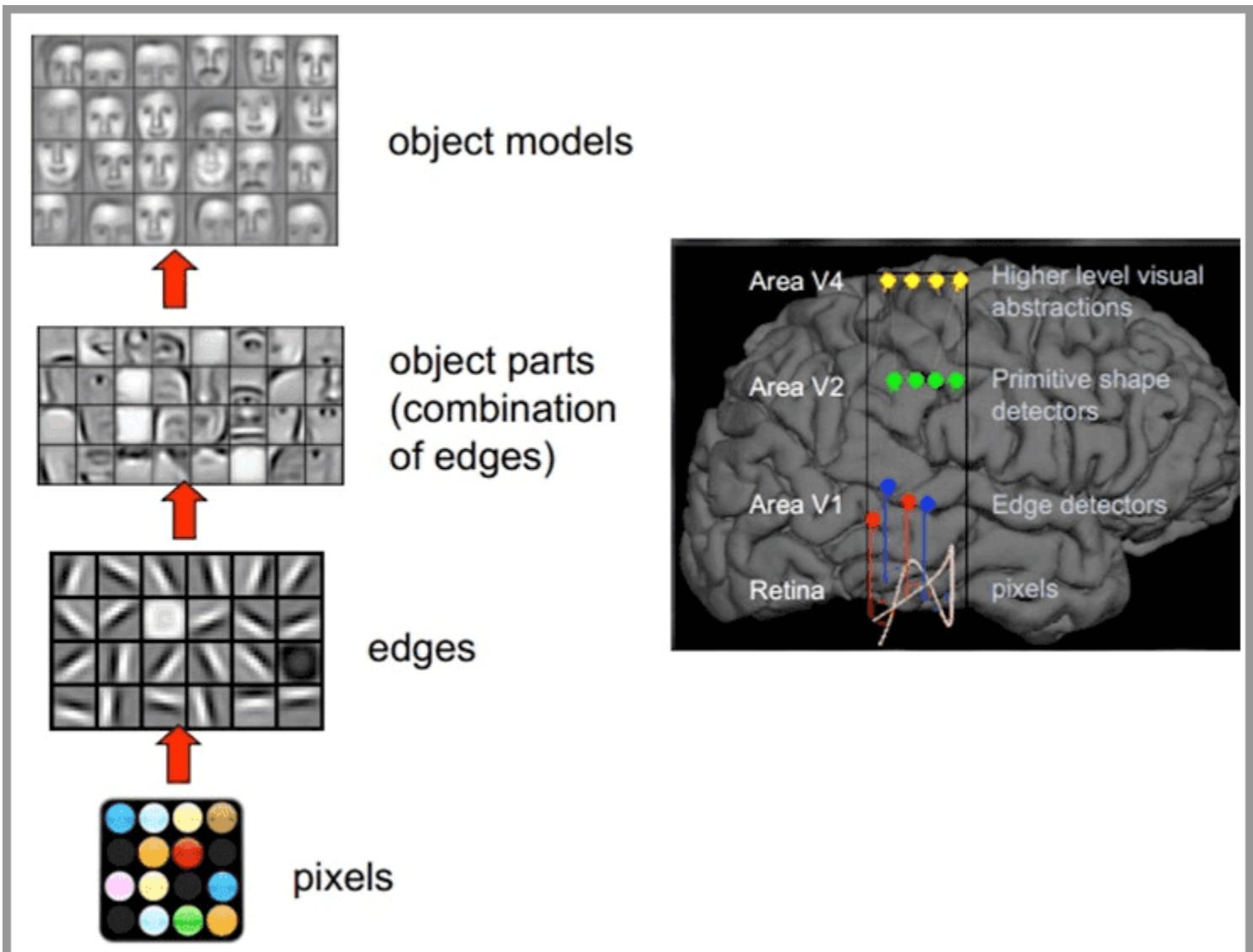
# Typical Convolutional Neural Network Architecture



# Typical Convolutional Neural Network Architecture



# Intuition



# Edge Detection

10	10	10	0	0	0
10	10	10	0	0	0
10	10	10	0	0	0
10	10	10	0	0	0
10	10	10	0	0	0
10	10	10	0	0	0



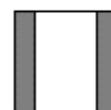
\*

1	0	-1
1	0	-1
1	0	-1



=

0	30	30	0
0	30	30	0
0	30	30	0
0	30	30	0



# Various Edge Detection Filters

Sobel

-1	0	1
-2	0	2
-1	0	1

45°

0	1	2
-1	0	1
-2	-1	0

Kirsch

-3	-3	5
-3	0	5
-3	-3	5

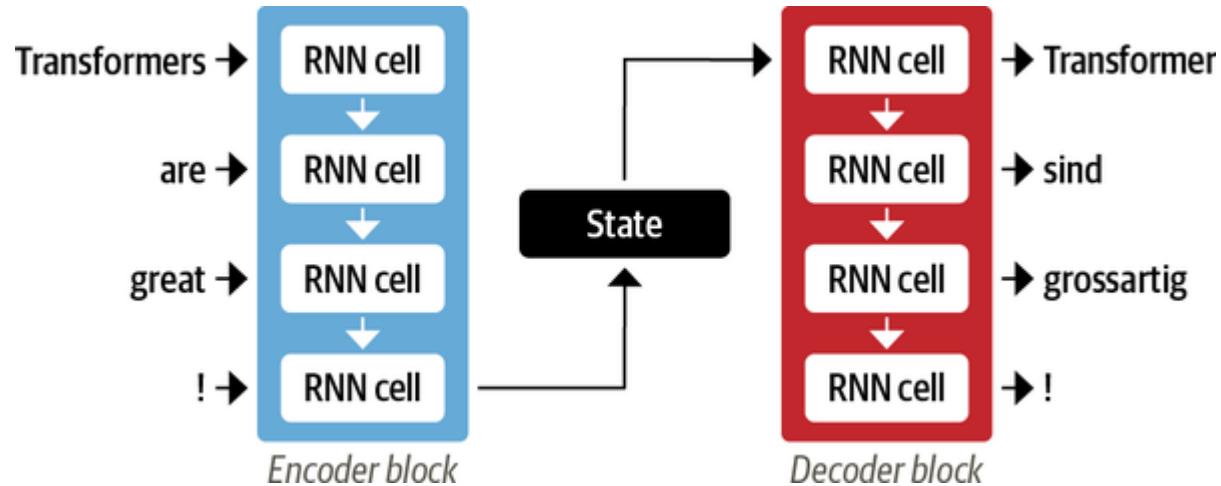
-3	5	5
-3	0	5
-3	-3	-3

Robinson

-1	0	1
-1	0	1
-1	0	1

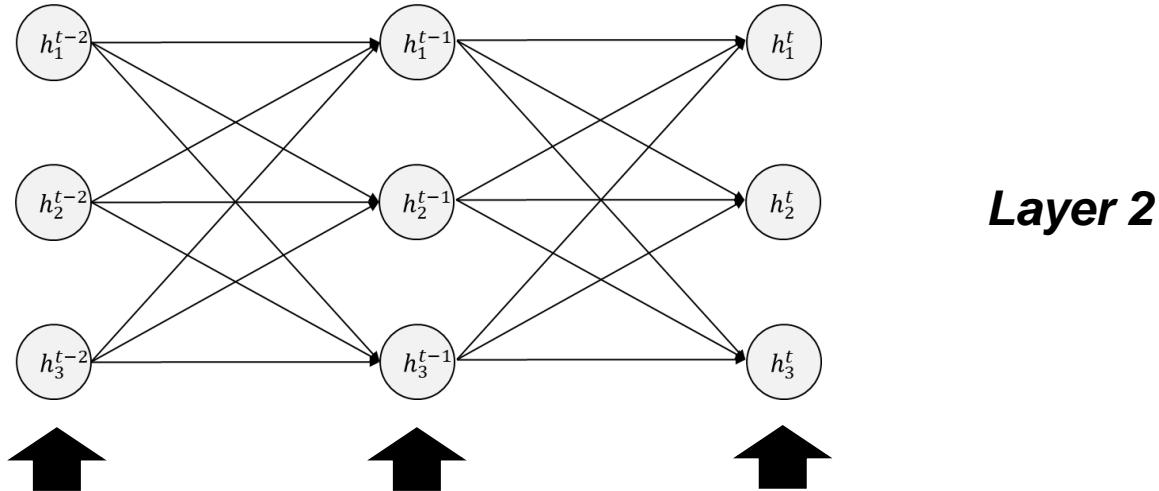
0	1	1
-1	0	1
-1	-1	0

# Evolution Of Transformers: Encoder-Decoder Model for Translation

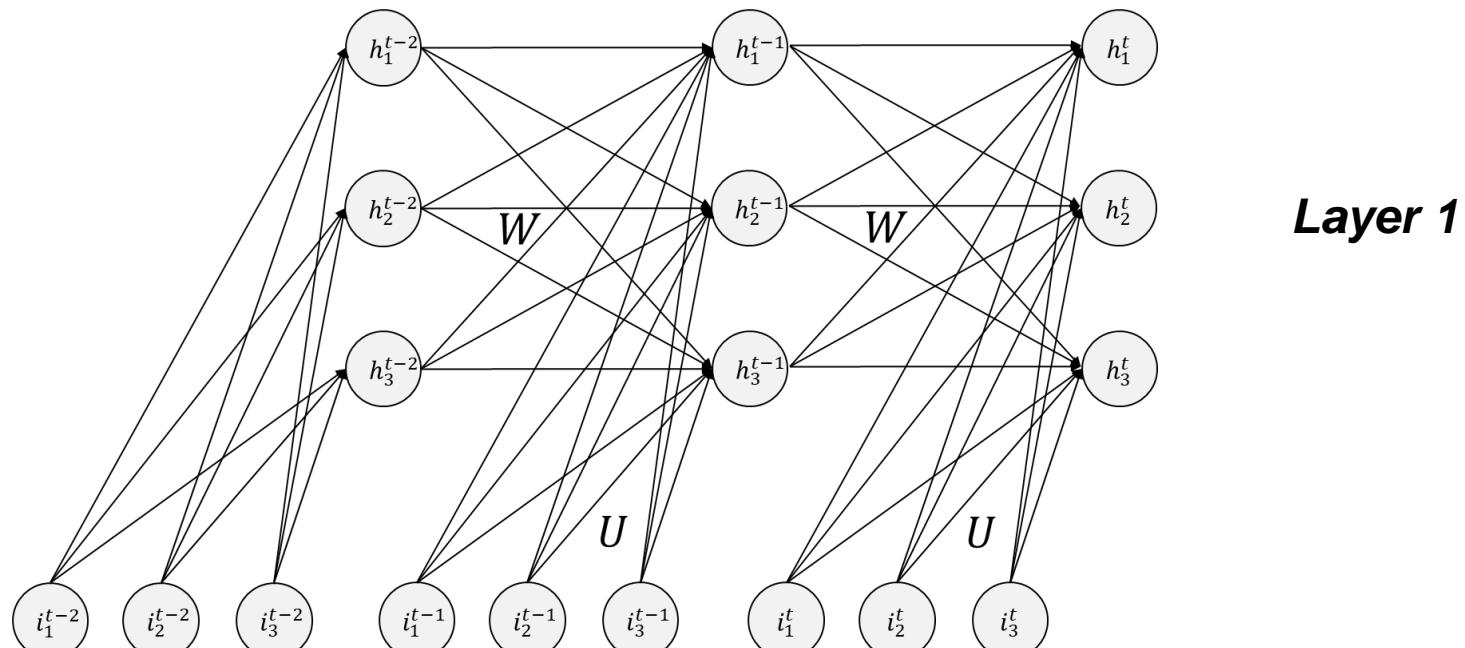


Information bottleneck: meaning of entire sequence in final hidden state  
(esp. problematic for long sequences)

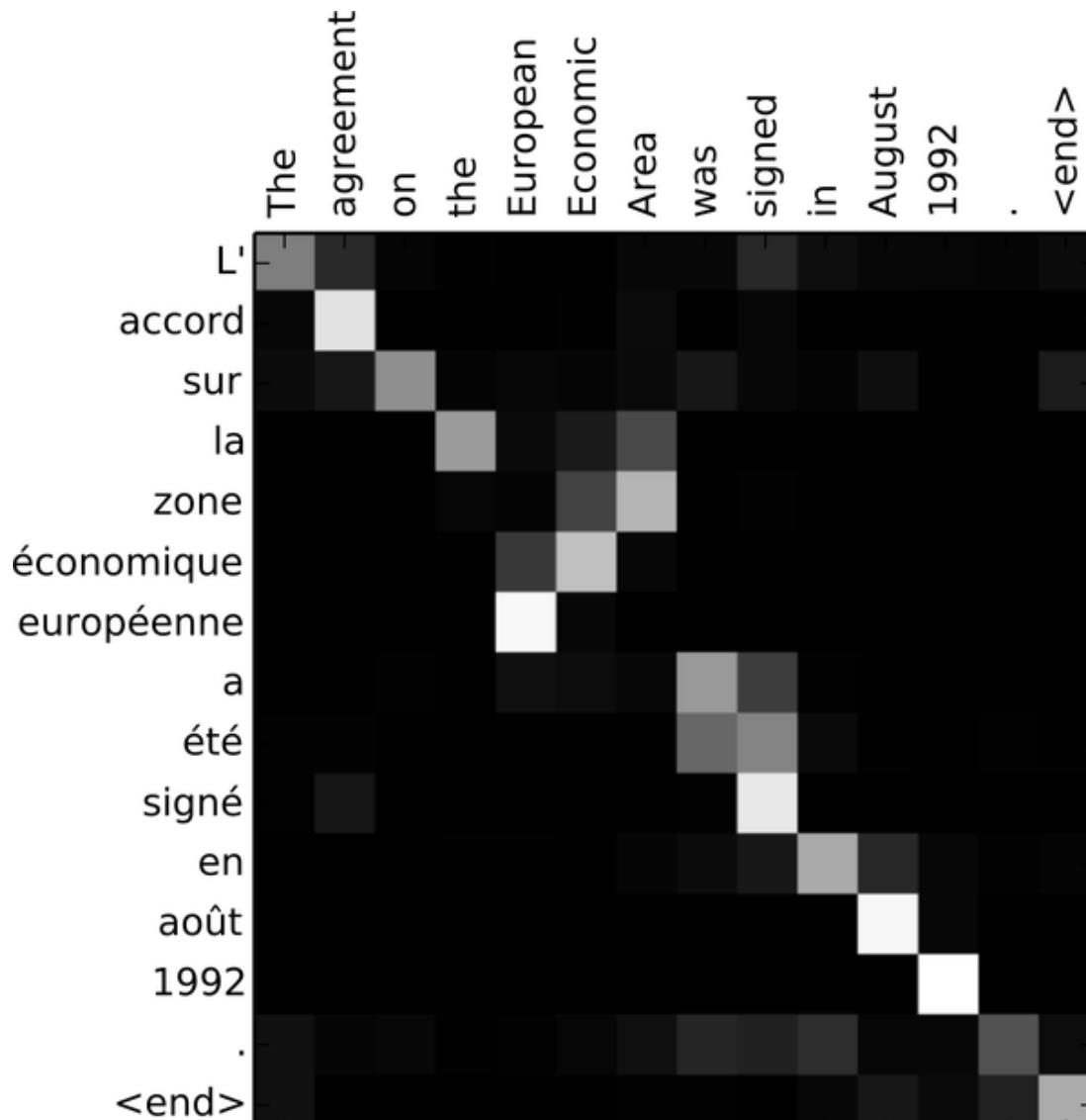
# Multiple Layers of RNNs



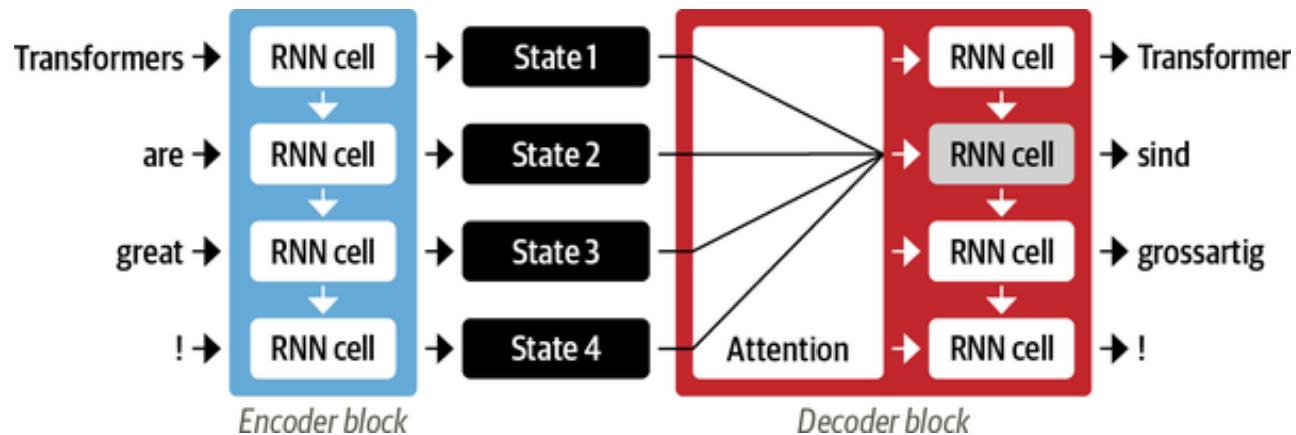
Output sequence of layer  
1 is input of layer 2



# Information Is Often In Different Orders

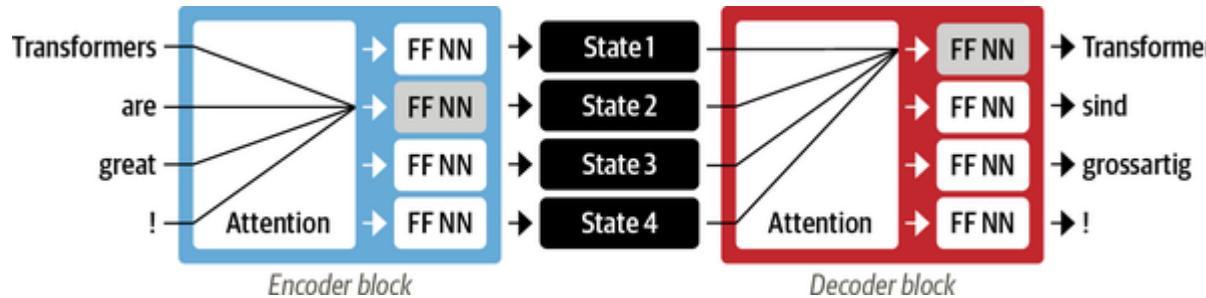


# Add Attention Layer

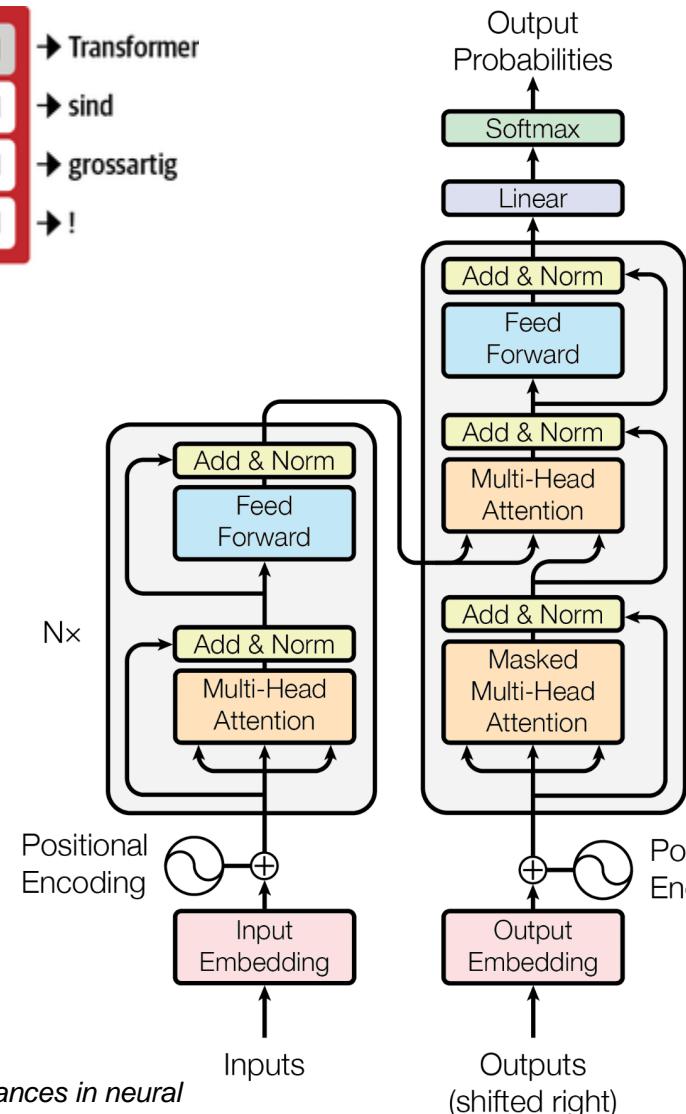


Computations are inherently sequential and cannot be parallelized across input sequence

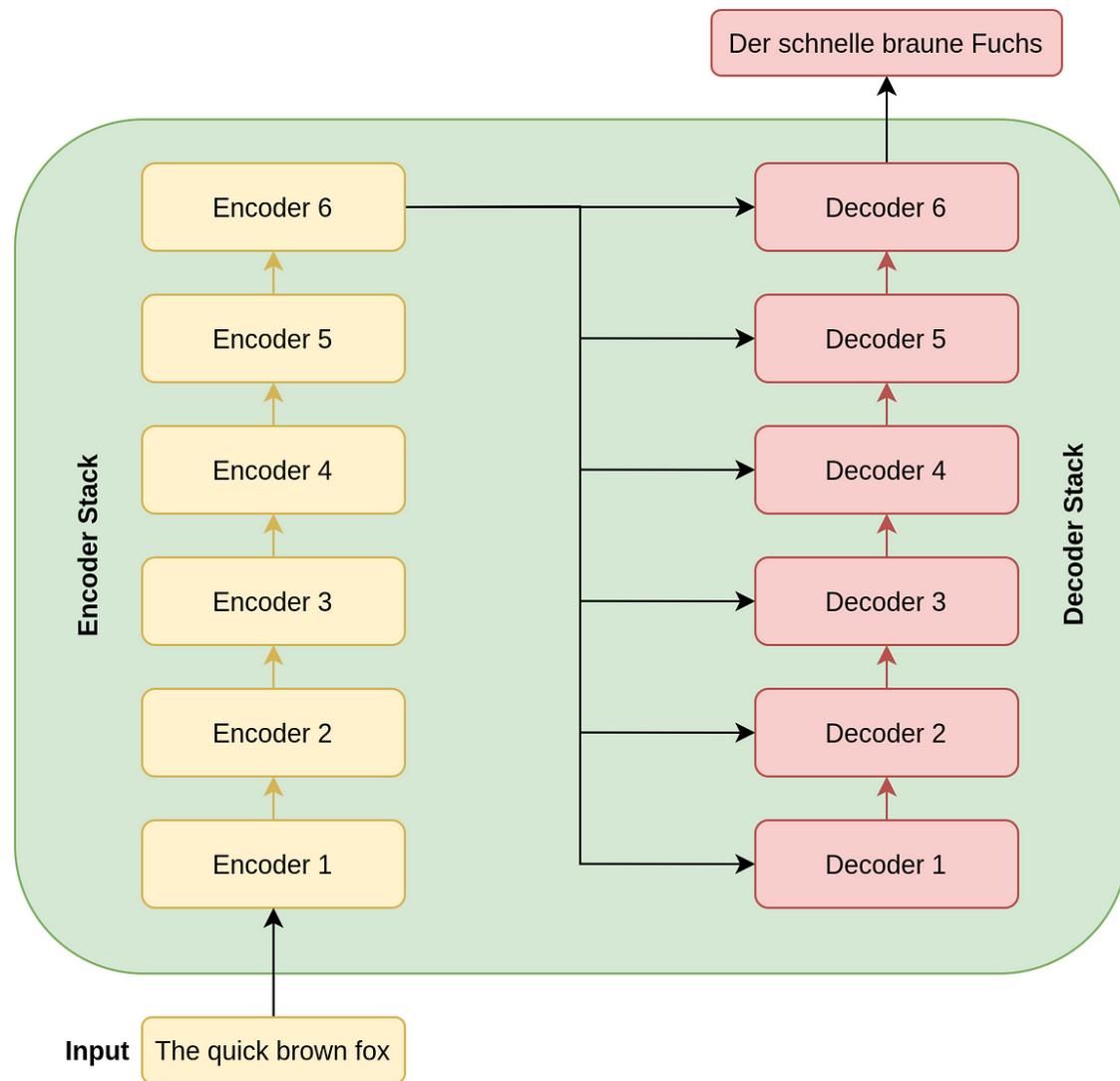
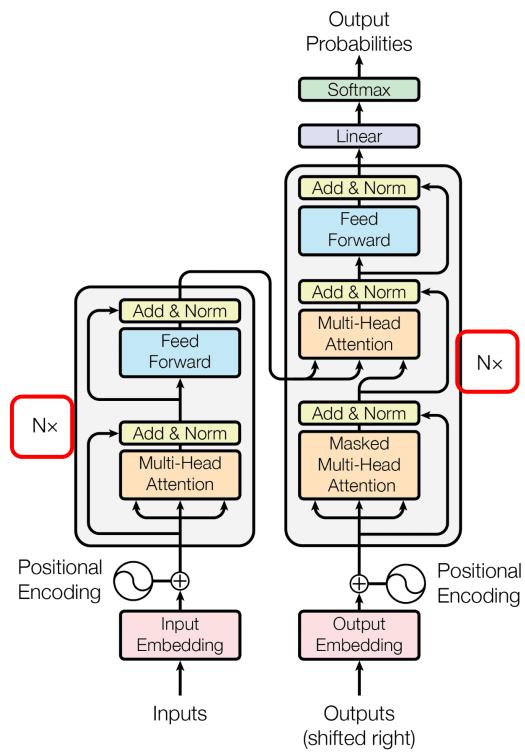
# Attention Is All You Need



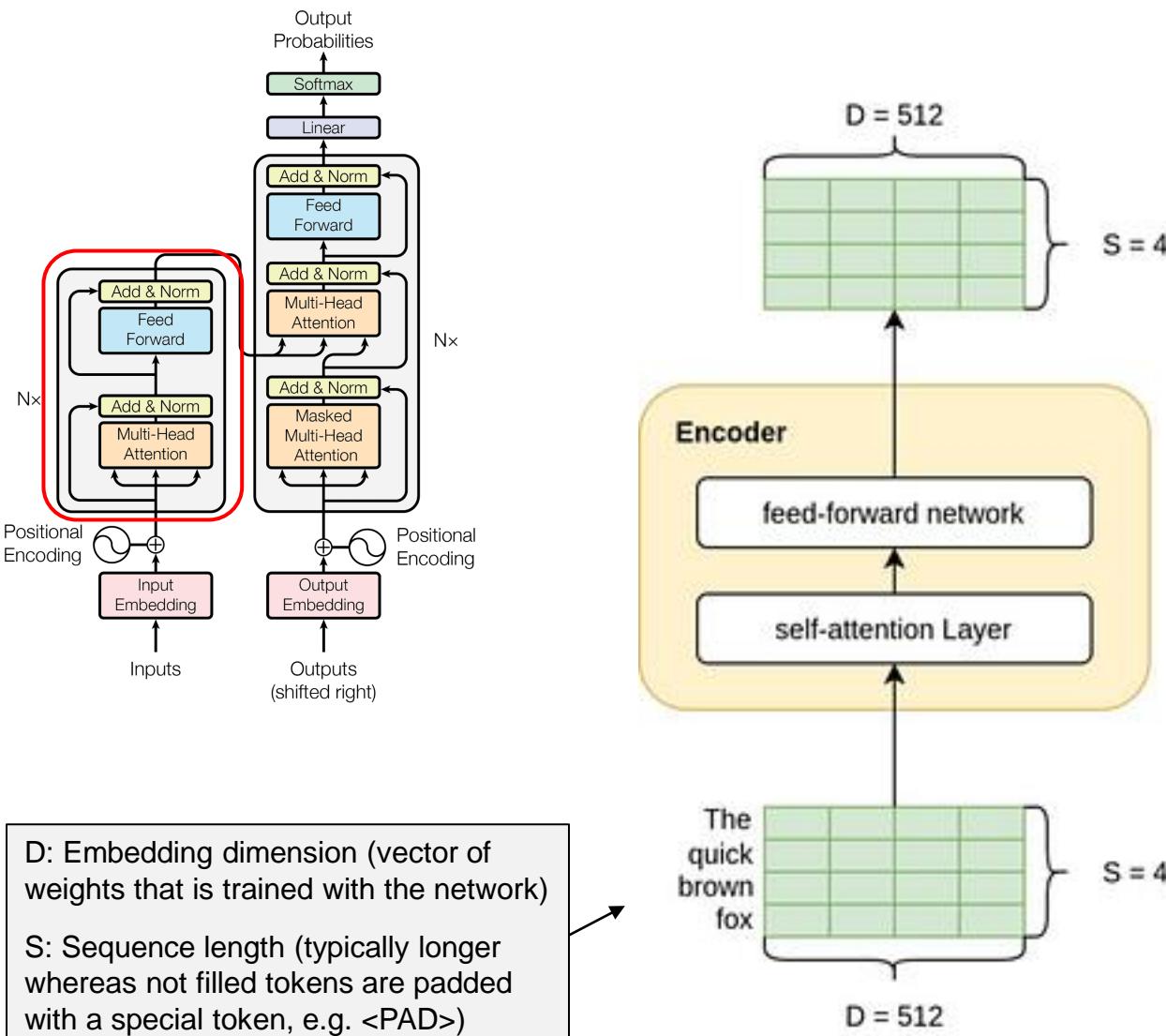
Tunstall, Lewis, Leandro Von Werra, and Thomas Wolf. *Natural language processing with transformers.* " O'Reilly Media, Inc.", 2022.



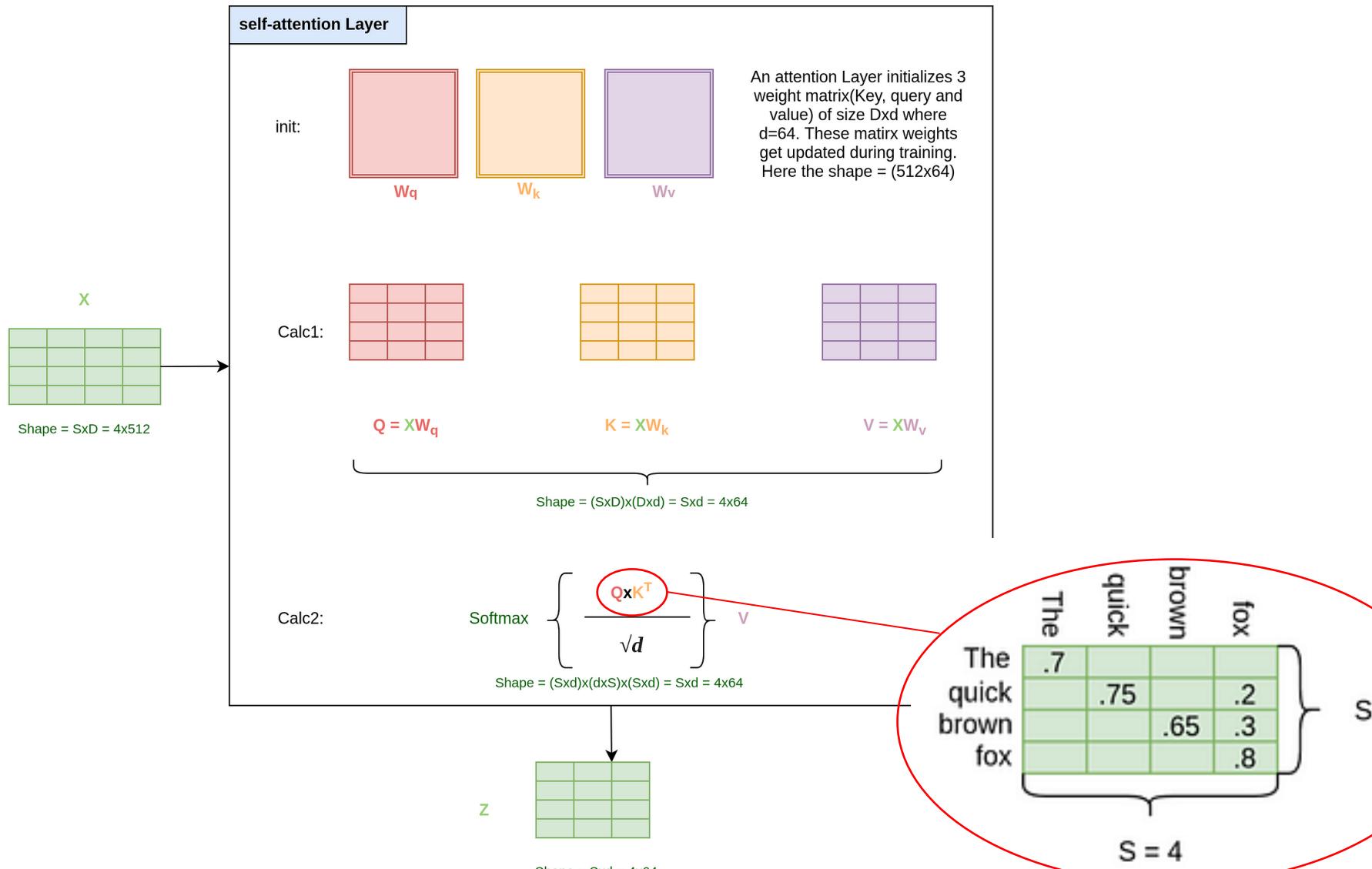
# Encoder-Decoder



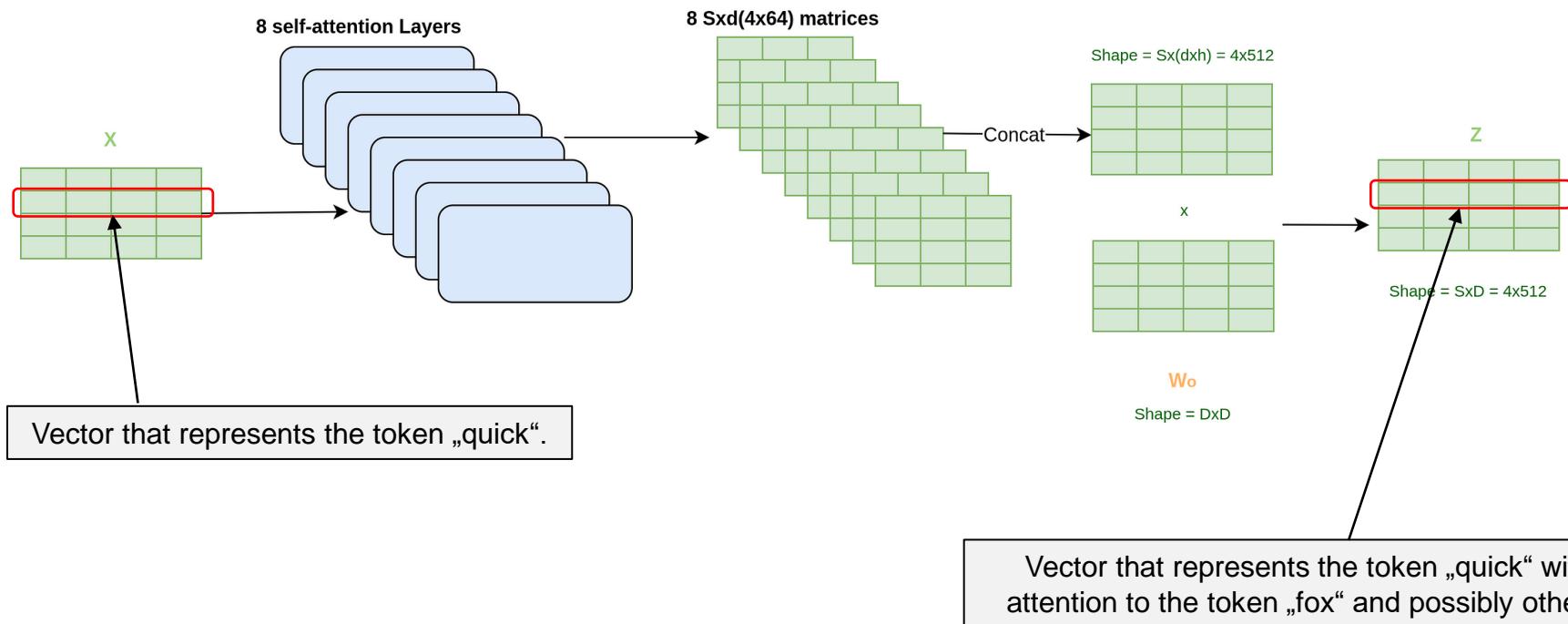
# Input For Encoder



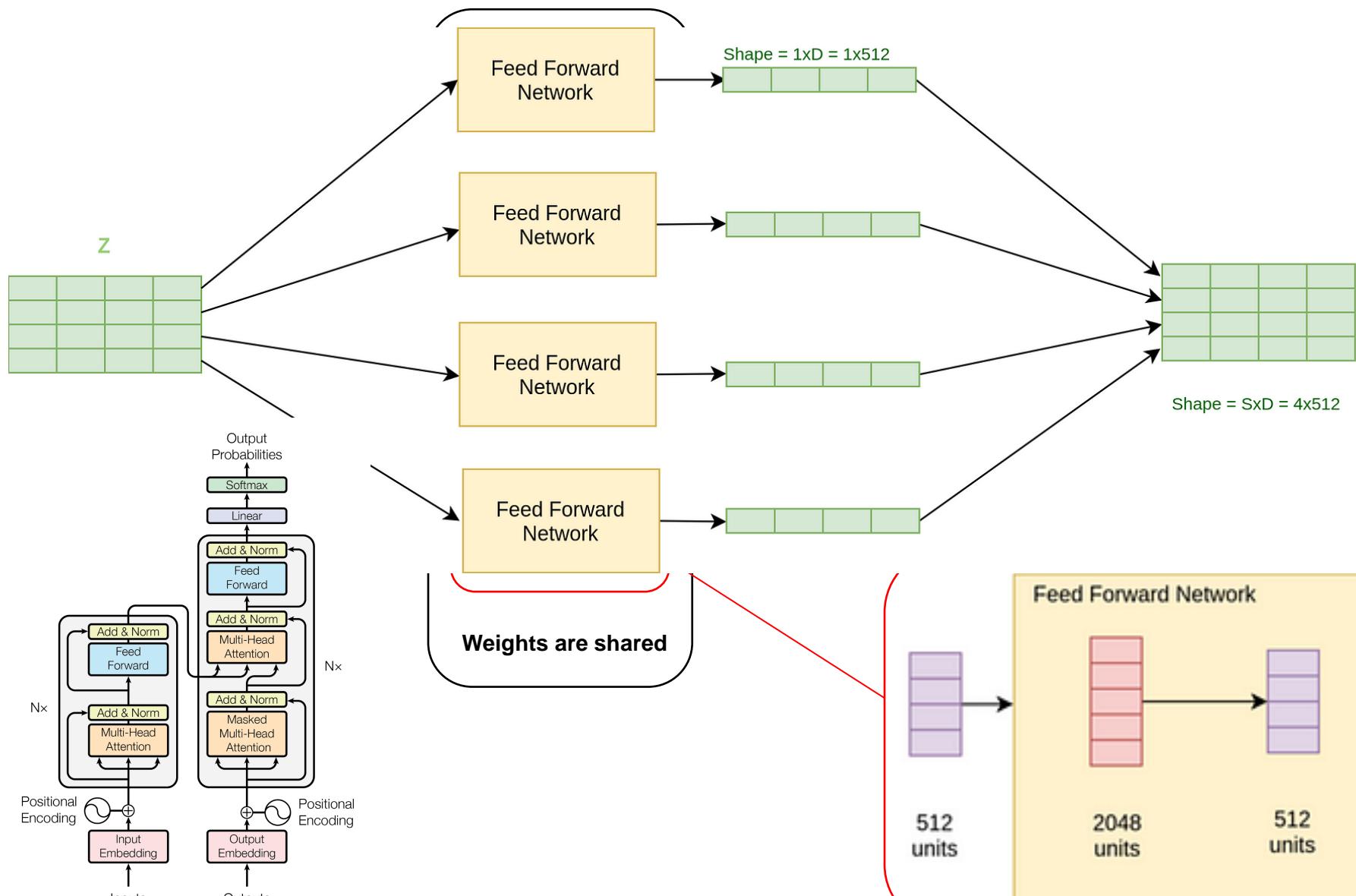
# Self-Attention Layer



# Multi-Head Self-Attention Layer

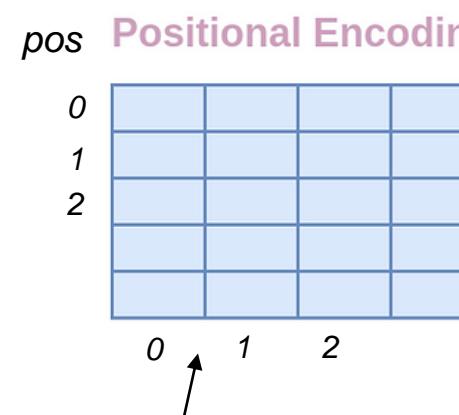
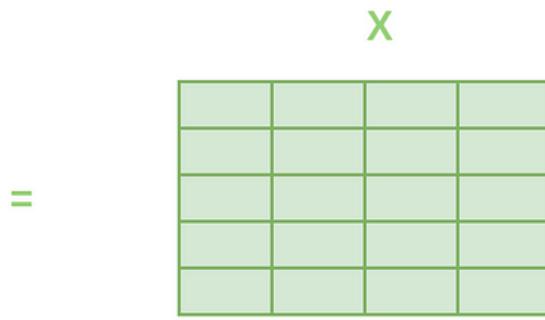
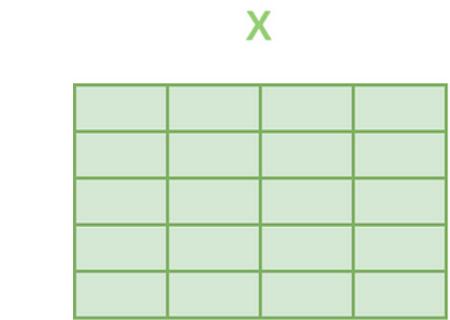


# Feed Forward Layer (Position-Wise)



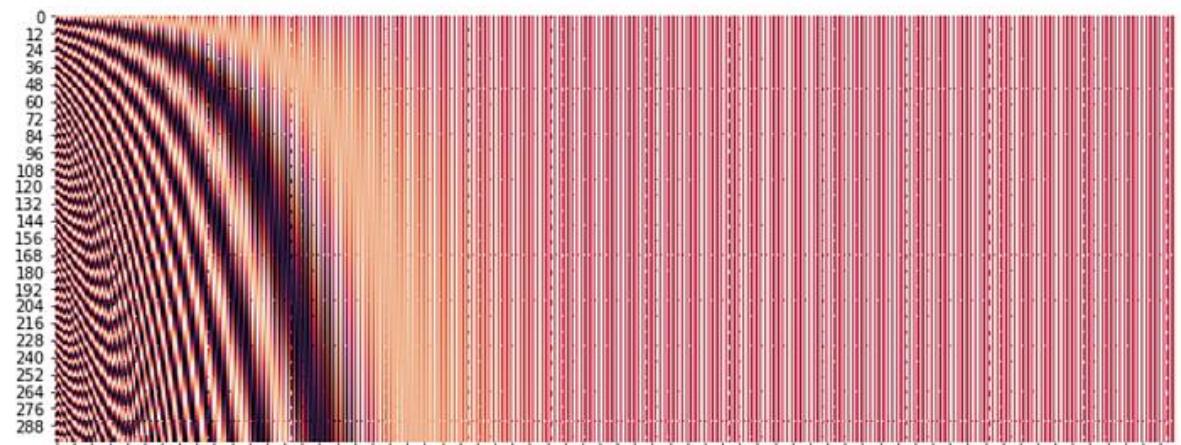
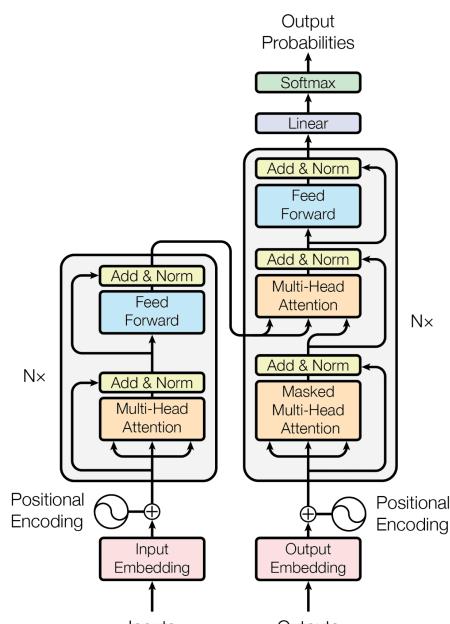
# Positional Encoding

<https://www.kdnuggets.com/2020/10/understanding-transformers-data-science.html>

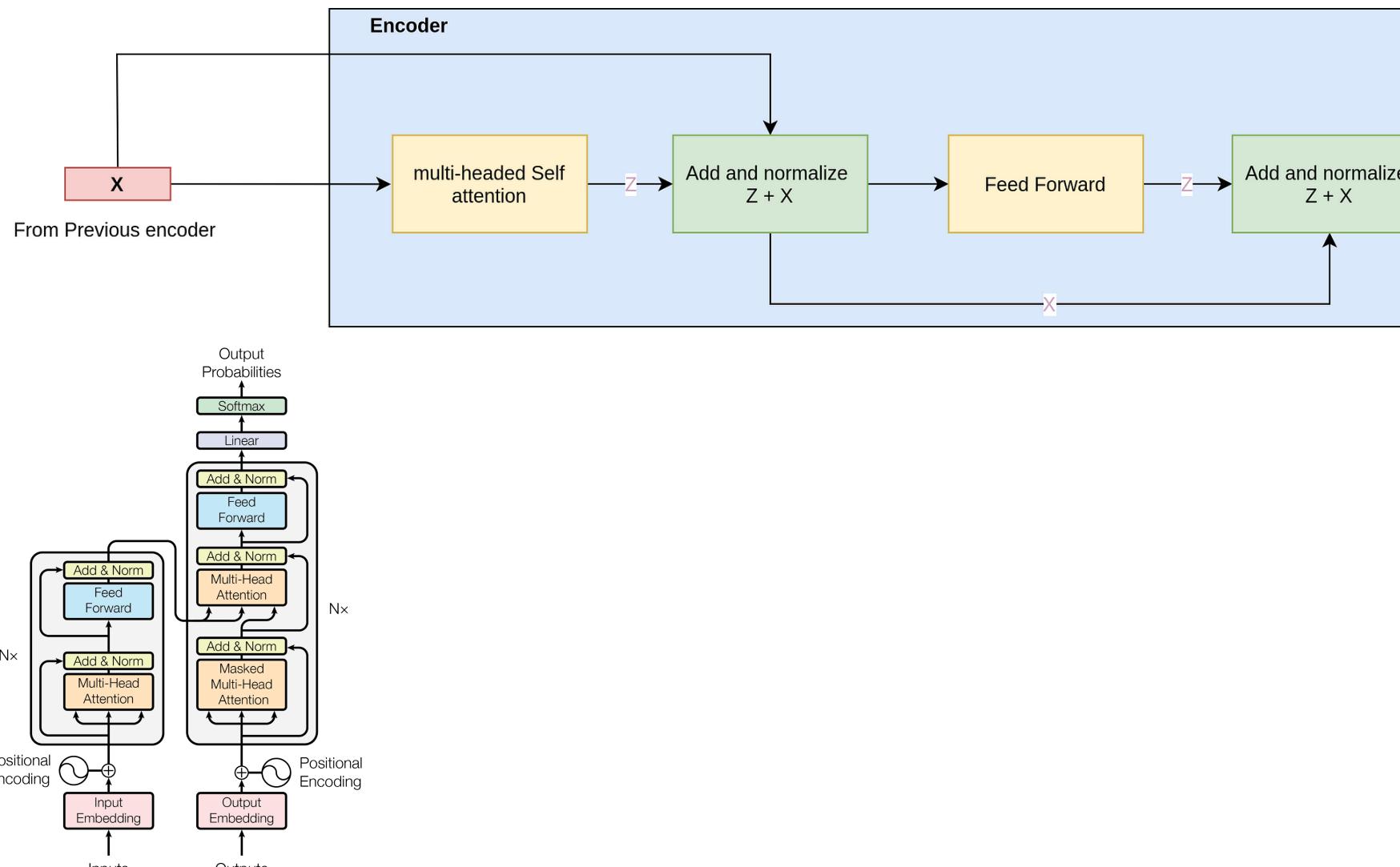


$$PE_{(pos,2i)} = \sin(pos/10000^{2i/D})$$

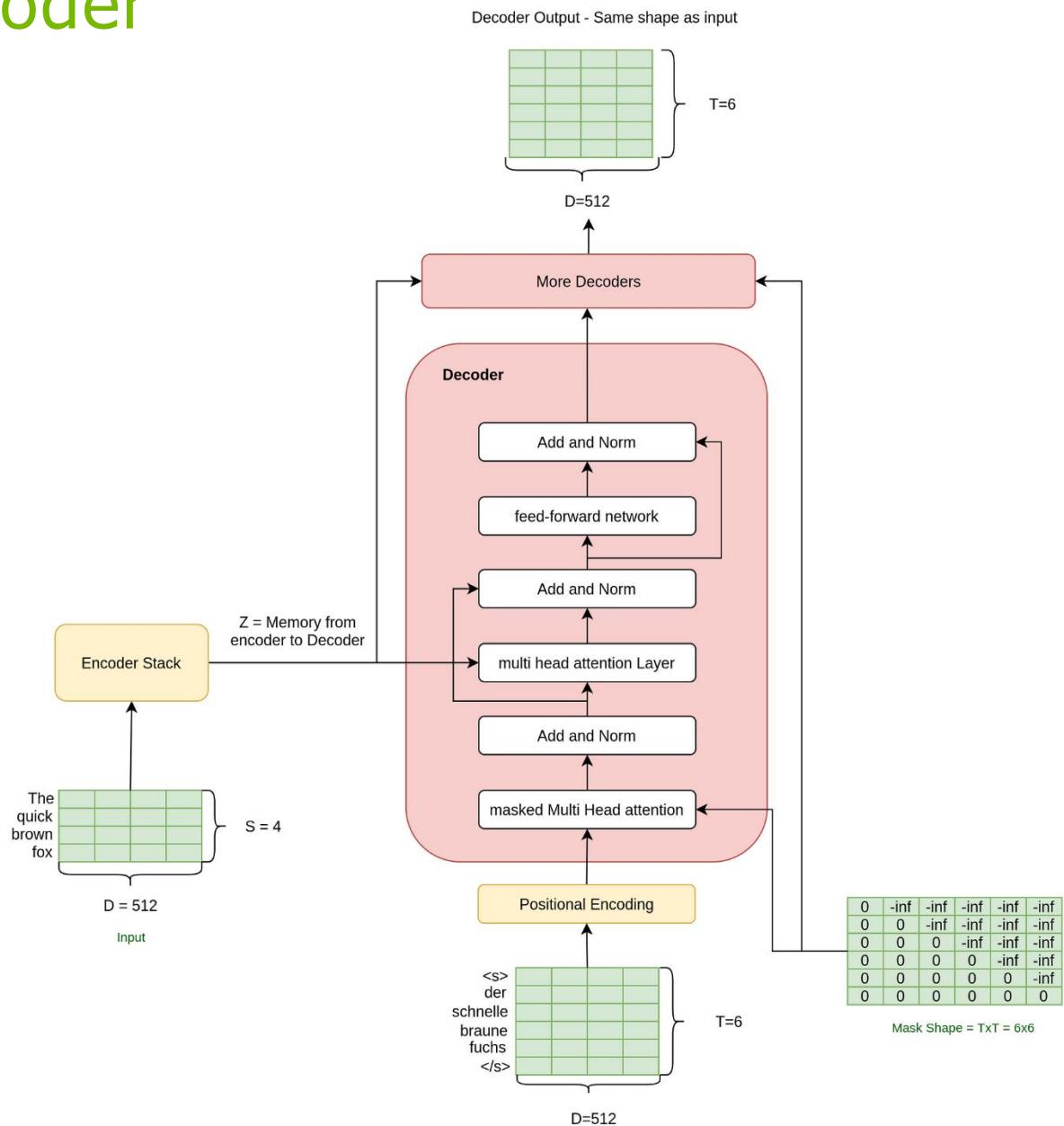
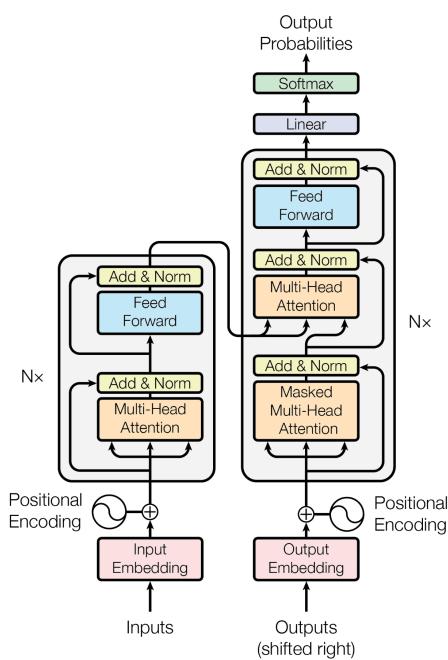
$$PE_{(pos,2i+1)} = \cos(pos/10000^{2i/D})$$



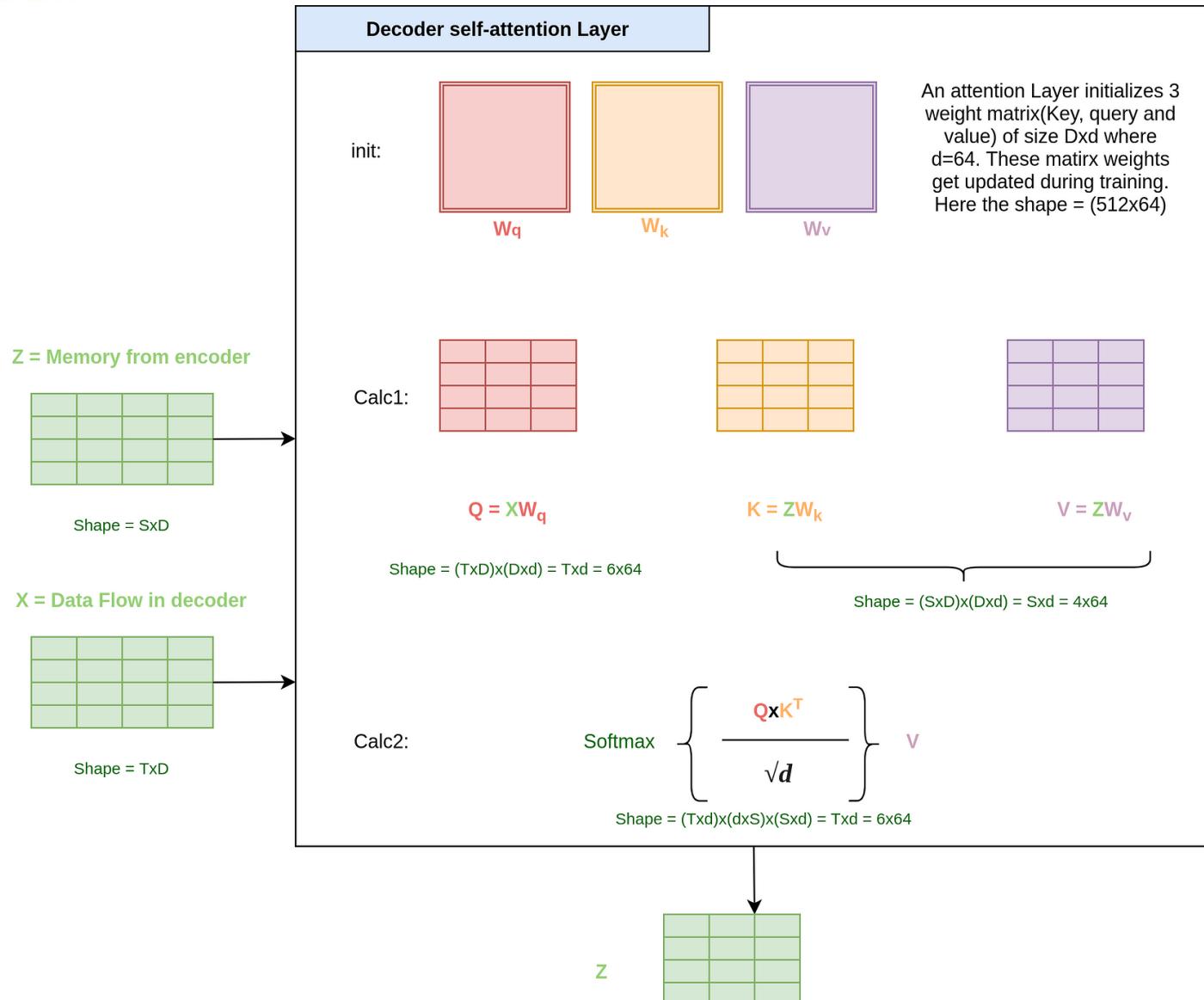
# Add & Norm (Residual Connections)



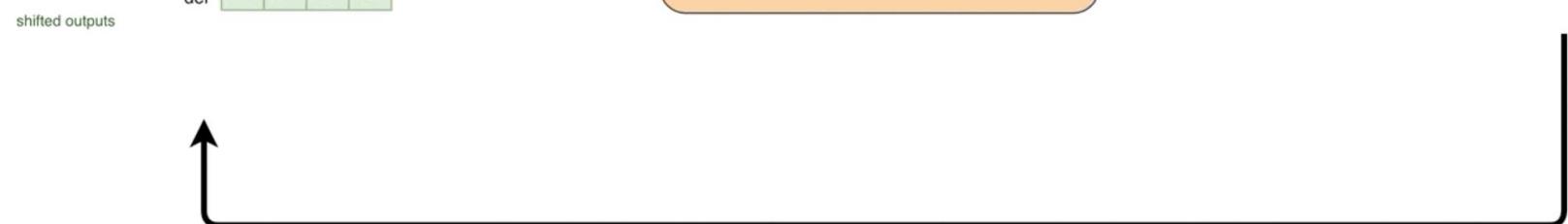
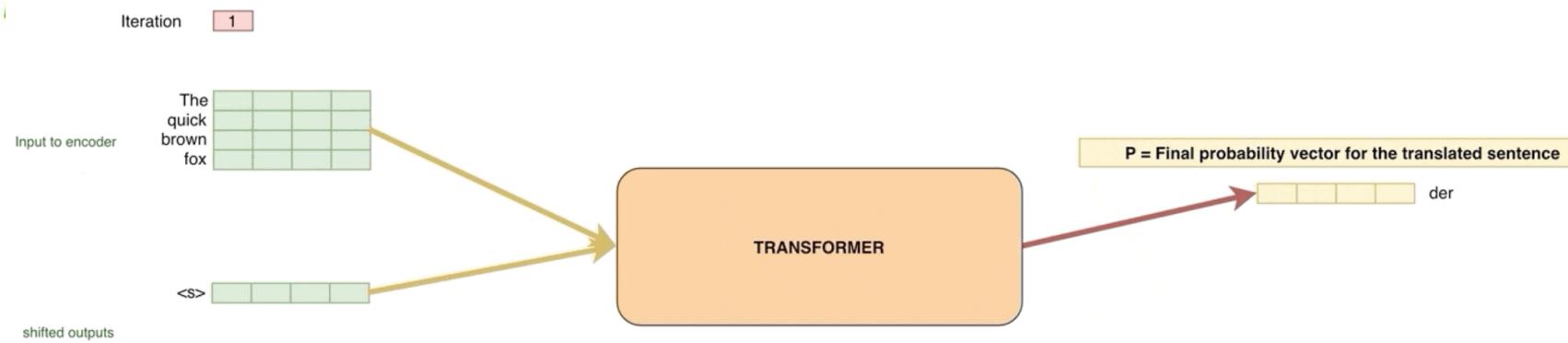
# Decoder



# Decoder Self-Attention Layer



# Predict Token By Token

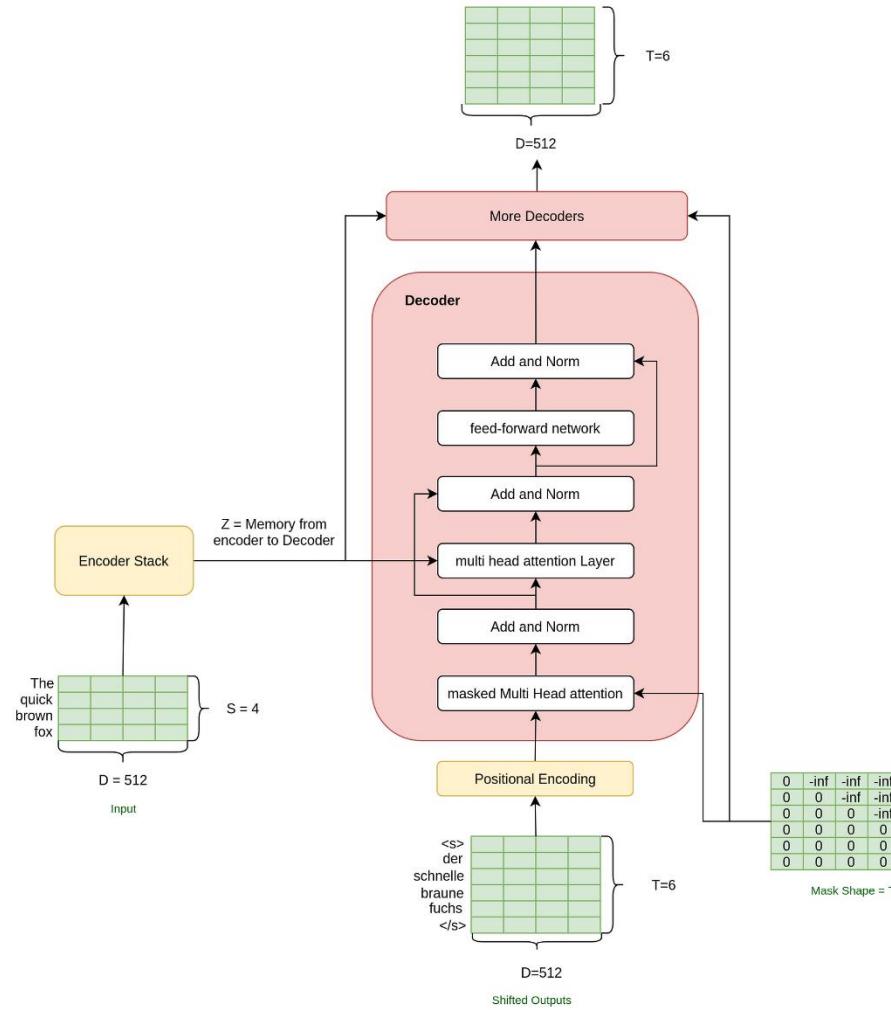


# Training

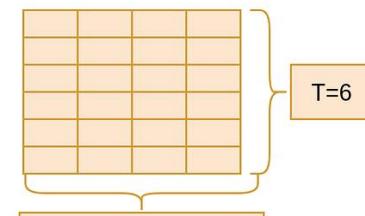
The Quick Brown Fox  
 < s > Der schnelle braune fuchs < /s >



Decoder Output - Same shape as input



P = Final probability vector for the translated sentence



# Masking

<S>					
der					
schnelle					
braune					
fuchs					
</S>					
	100	20	80		
		100	80		

+

0	-inf	-inf	-inf	-inf	-inf
0	0	-inf	-inf	-inf	-inf
0	0	0	-inf	-inf	-inf
0	0	0	0	-inf	-inf
0	0	0	0	0	-inf
0	0	0	0	0	0

||

<S>	-inf	-inf	-inf		
der	inf	inf	inf		
schnelle	100	-inf	-inf		
braune		100	-inf		
fuchs			100		
</S>					

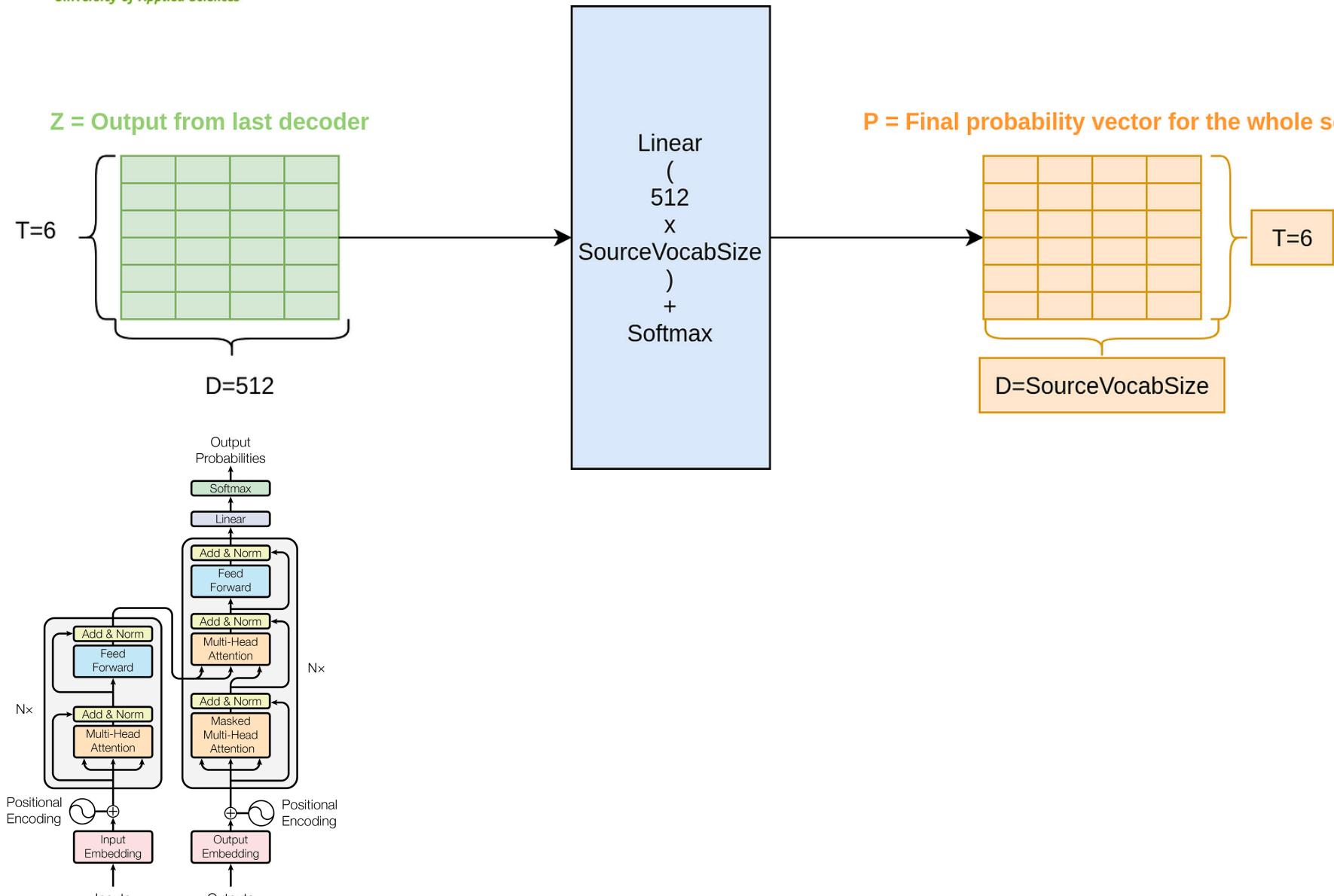
Mask Shape = TxT = 6x6

$$Q \times K^T / \sqrt{d}$$

$$Q \times K^T / \sqrt{d}$$

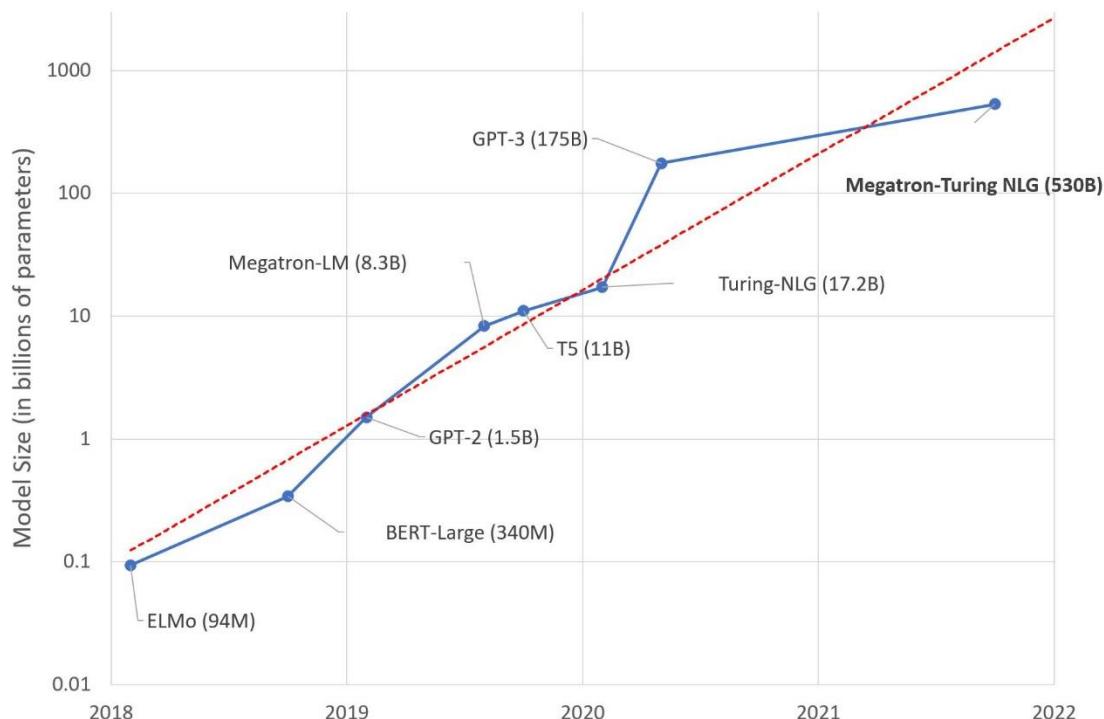
	0	0	0	0	0
		0	0	0	0
		.75	0	0	0
			.85	0	0
				0	

# Linear Layer



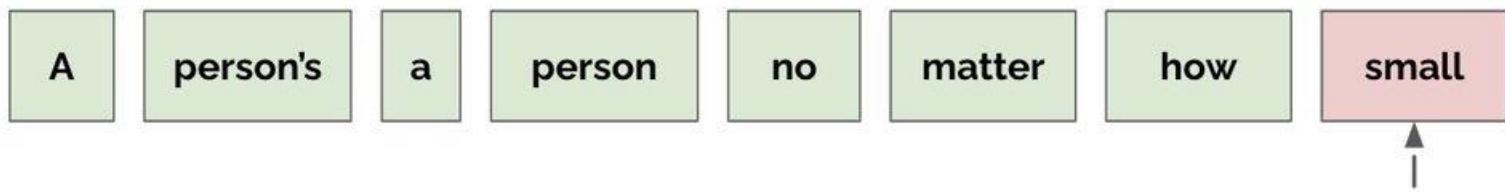
# Evolution Of Transformer Architectures

- Different families of transformer architectures evolved:
  - Encoder branch: BERT, DistilBERT, RoBERTa, XLM, XLM-R, ALBERT, ELECTRA, DeBERTa
  - Decoder branch: GPT, GPT-2, GPT-3, GPT-Neo
  - Encoder-decoder branch: T5, BART, M2M-100, BigBird

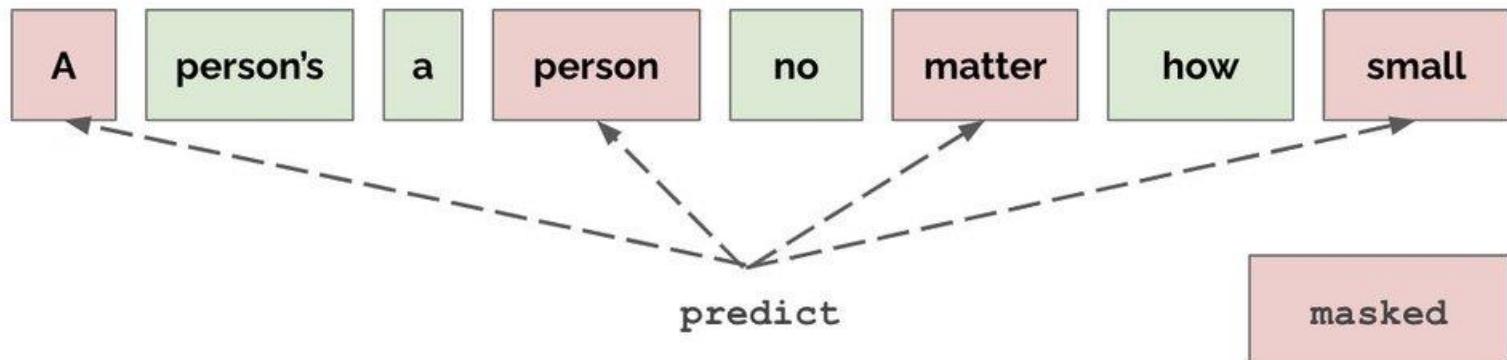


# 2 Ways to Pretrain LLMs Generically (Very Large Corpora)

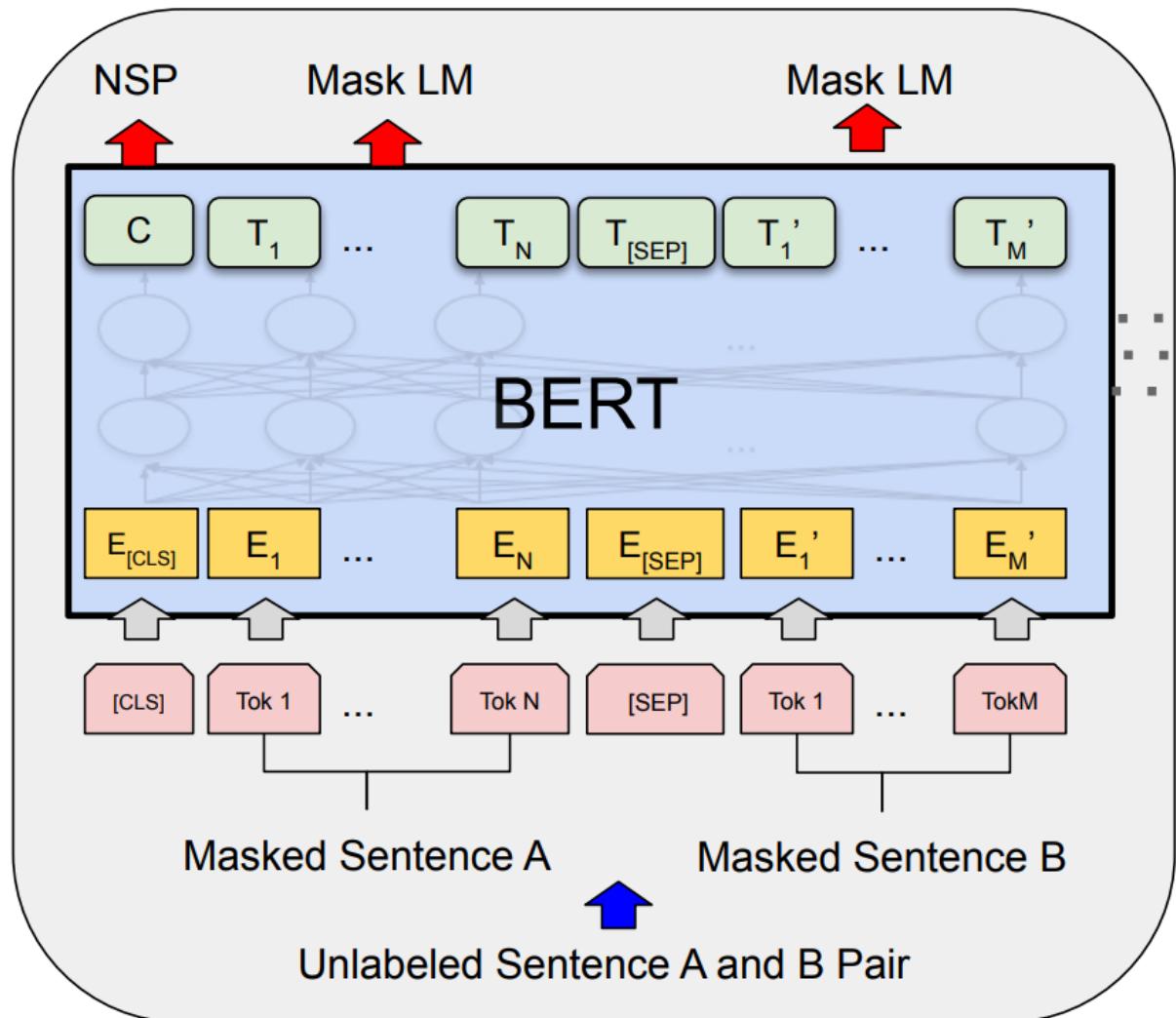
## GPT - causal mask



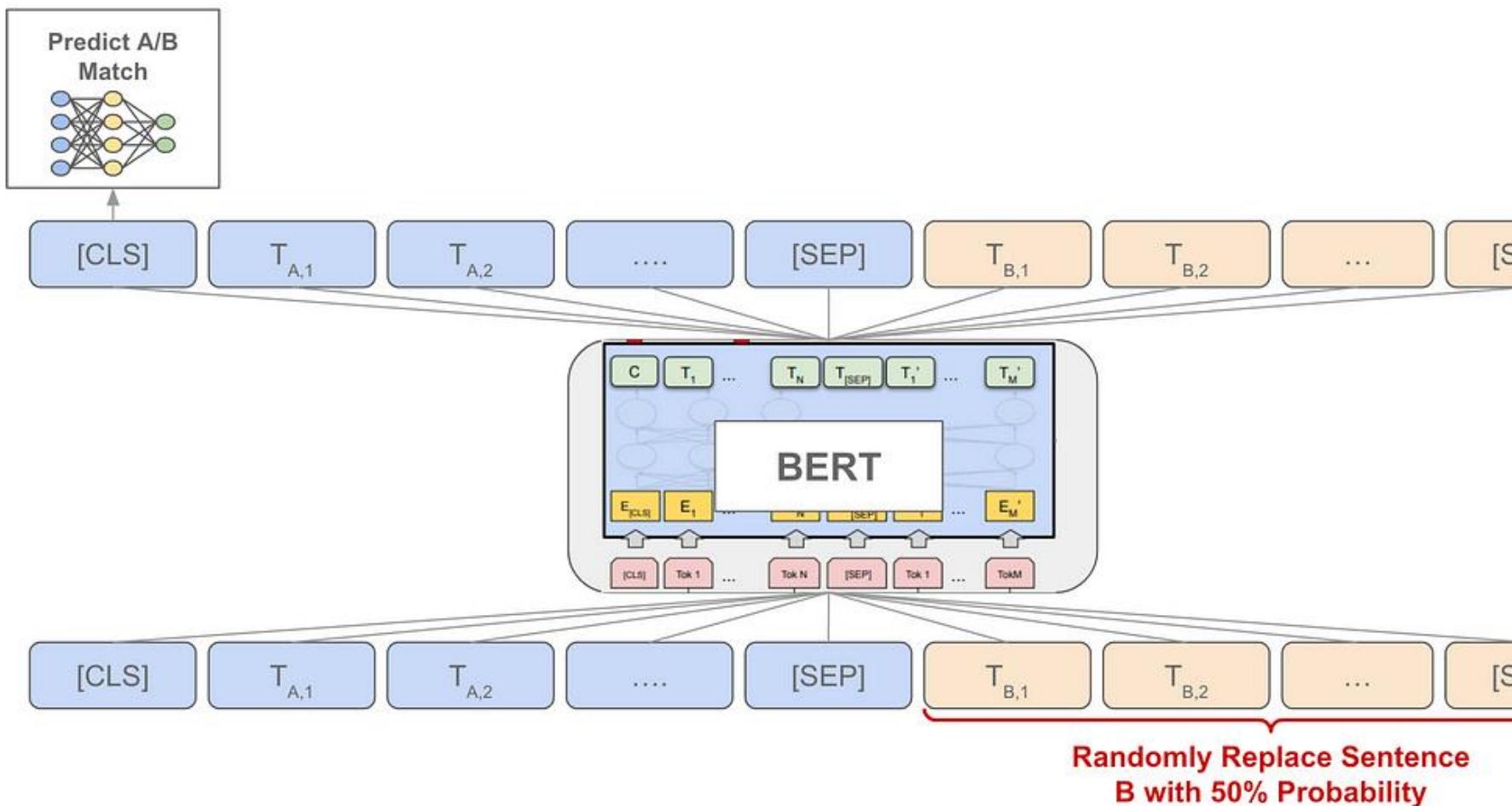
## BERT - random mask



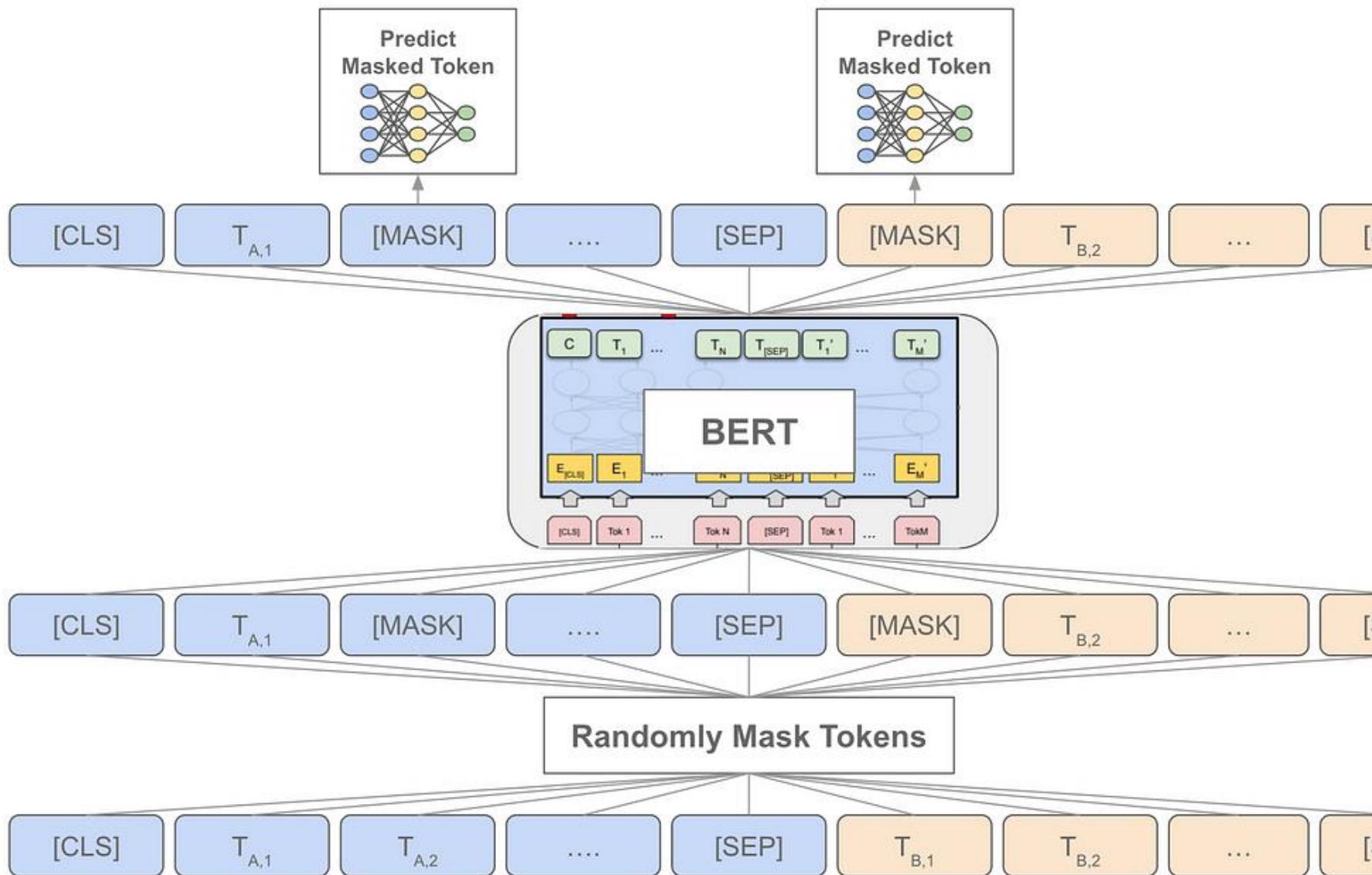
# BERT Pre-Training



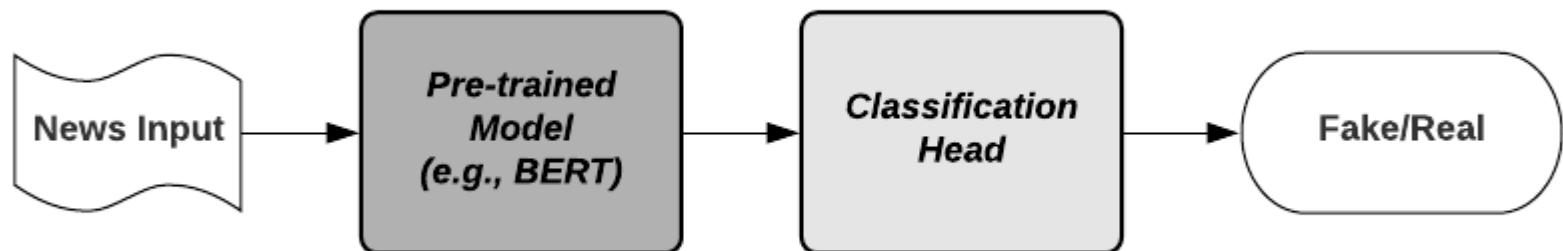
# BERT Pre-Training: Sentence Matching



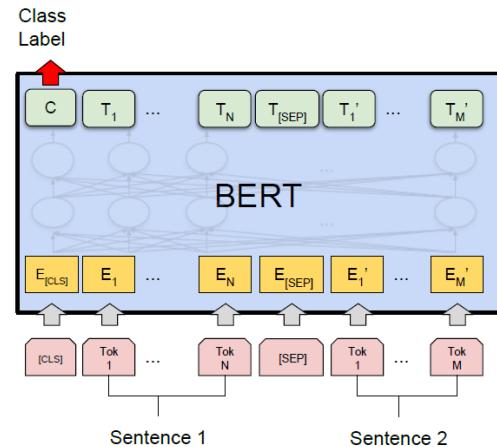
# BERT Pre-Training: Masked Token Prediction



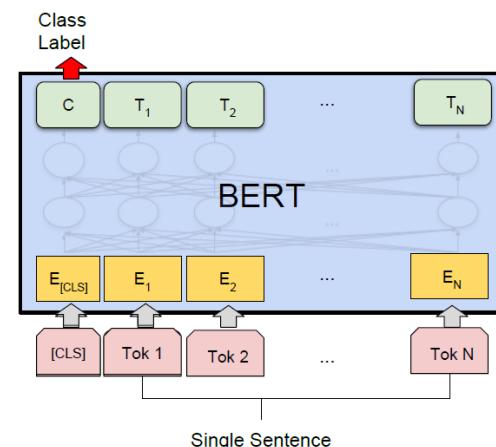
# Fine-Tuning For Downstream Tasks



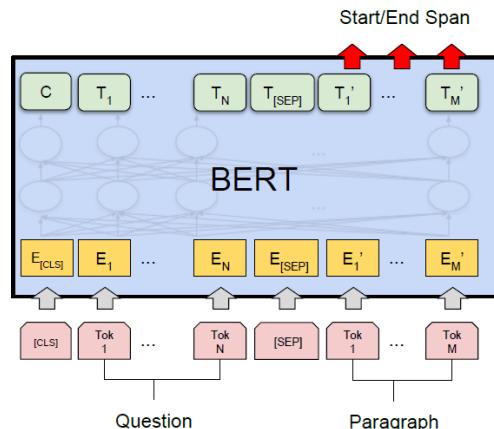
# Fine-Tuning For Downstream Tasks



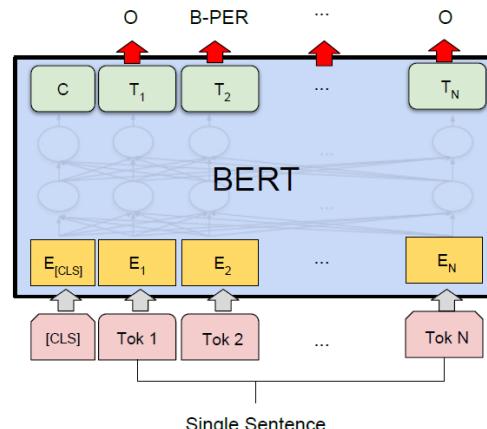
(a) Sentence Pair Classification Tasks:  
MNLI, QQP, QNLI, STS-B, MRPC,  
RTE, SWAG



(b) Single Sentence Classification Tasks:  
SST-2, CoLA

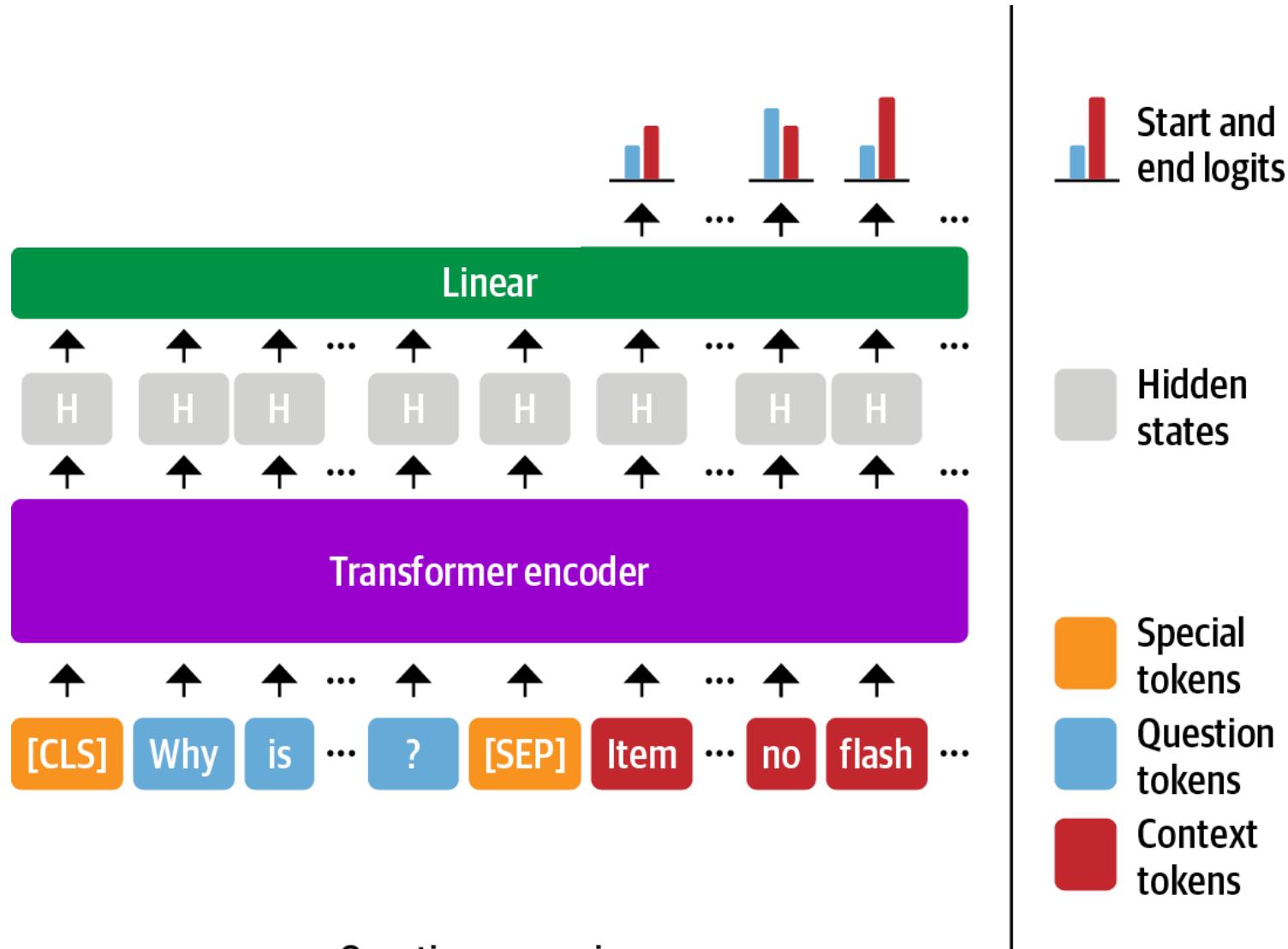


(c) Question Answering Tasks:  
SQuAD v1.1



(d) Single Sentence Tagging Tasks:  
CoNLL-2003 NER

# Extractive Question Answering



# Sliding Window For Long Documents

Why is the camera of poor quality? Item like the picture, fast deliver 3 days well packed, g  
quality for the price. The camera is decent (as phone cameras go). There is no flash though.



[CLS] [REDACTED] [SEP] [REDACTED] [S]

Stride

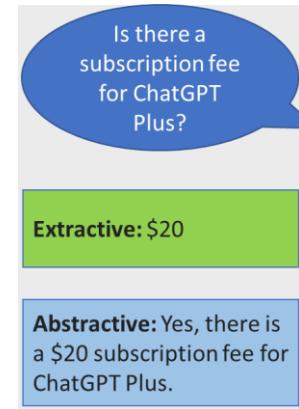
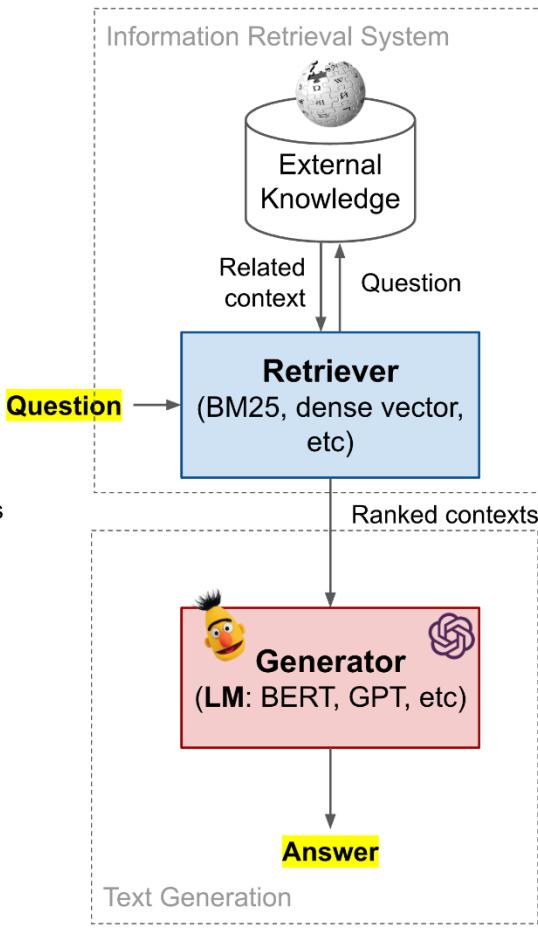
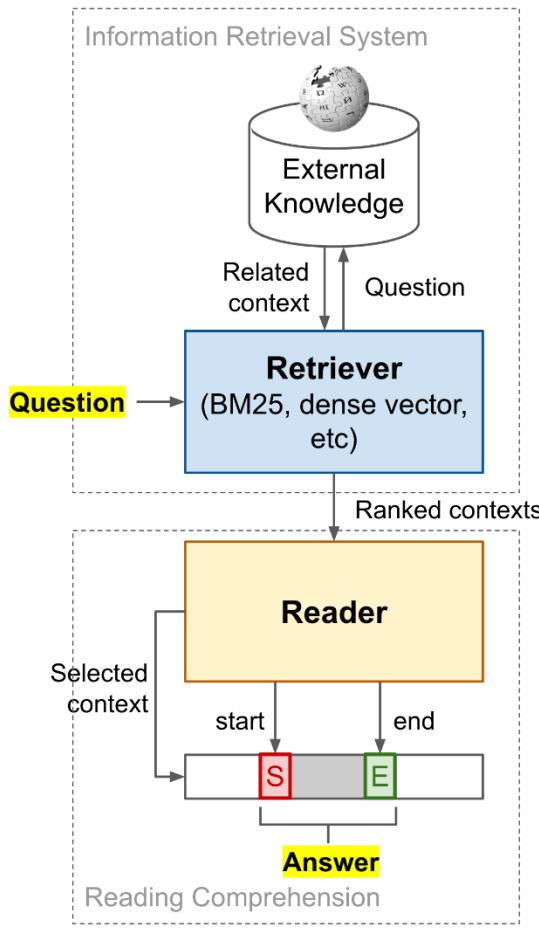


Why is the camera of poor quality? Item like the picture, fast deliver 3 days well packed, g  
quality for the price. The camera is decent (as phone cameras go). There is no flash though.

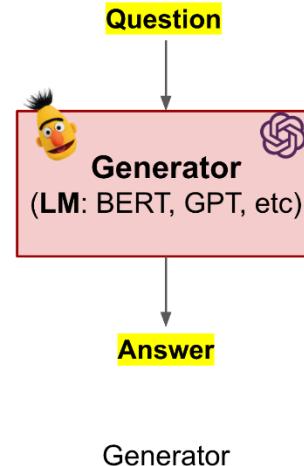


[CLS] [REDACTED] [SEP] [REDACTED] [S]

# From Extractive To Abstractive Q&A

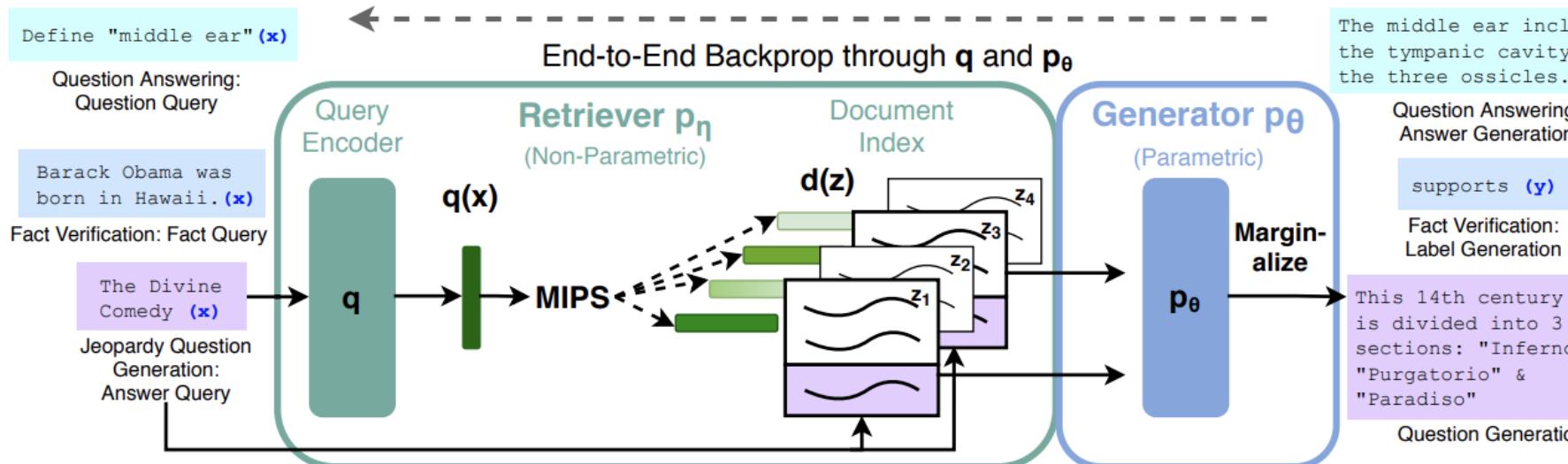


OpenAI is launching a premium and version of ChatGPT. The free app will be available. But it is liable to go offline during busy periods – and, during those, those who have paid its monthly fee will have priority access. That is just one of the offered in return for the \$20 subscription fee for "ChatGPT Plus".



Generator

# RAG (Retrieval Augment Generation) Architecture for QA



## 2.2 Retriever: DPR

The retrieval component  $p_\eta(z|x)$  is based on DPR [26]. DPR follows a bi-encoder architecture:

$$p_\eta(z|x) \propto \exp(\mathbf{d}(z)^\top \mathbf{q}(x)) \quad \mathbf{d}(z) = \text{BERT}_d(z), \quad \mathbf{q}(x) = \text{BERT}_q(x)$$

where  $\mathbf{d}(z)$  is a dense representation of a document produced by a  $\text{BERT}_{\text{BASE}}$  *document encoder* [8], and  $\mathbf{q}(x)$  a query representation produced by a *query encoder*, also based on  $\text{BERT}_{\text{BASE}}$ . Calculating top-k( $p_\eta(\cdot|x)$ ), the list of  $k$  documents  $z$  with highest prior probability  $p_\eta(z|x)$ , is a Maximum Inner Product Search (MIPS) problem, which can be approximately solved in sub-linear time [23]. We use a pre-trained bi-encoder from DPR to initialize our retriever and to build the document index. This retriever was trained to retrieve documents which contain answers to TriviaQA [24] questions and Natural Questions [29]. We refer to the document index as the *non-parametric memory*.

## 2.3 Generator: BART

The generator component  $p_\theta(y_i|x, z, y_{1:i-1})$  could be modelled using any encoder-decoder. We use BART-large [32], a pre-trained seq2seq transformer [58] with 400M parameters. To combine the input  $x$  with the retrieved content  $z$  when generating from BART, we simply concatenate them. BART was pre-trained using a denoising objective and a variety of different noising functions. It has obtained state-of-the-art results on a diverse set of generation tasks and outperforms comparably-sized T5

# In-Context Learning

The three settings we explore for in-context learning

## Zero-shot

The model predicts the answer given only a natural language description of the task. No gradient updates are performed.



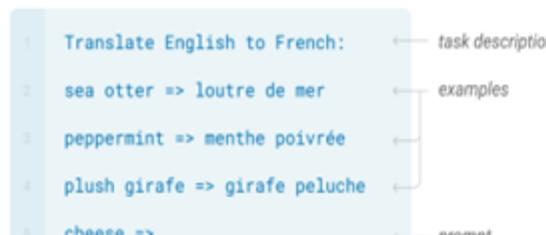
## One-shot

In addition to the task description, the model sees a single example of the task. No gradient updates are performed.



## Few-shot

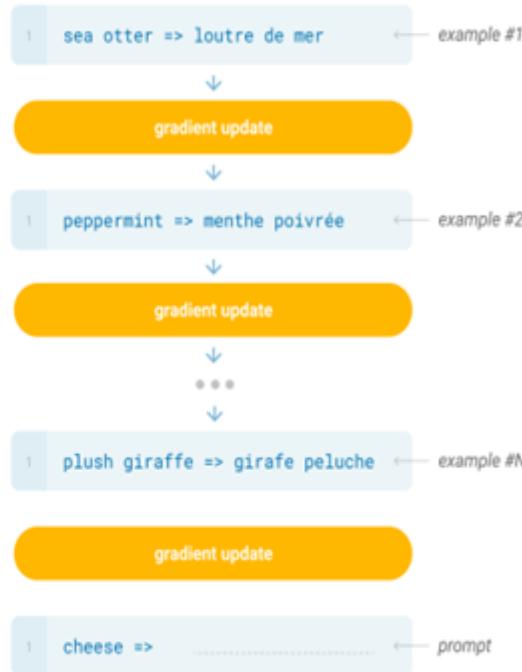
In addition to the task description, the model sees a few examples of the task. No gradient updates are performed.



Traditional fine-tuning (not used for GPT-3)

## Fine-tuning

The model is trained via repeated gradient updates using a large corpus of example tasks.



For abstractive Q&A, just provide example data (context, question, answer) such as:

**Context:** Matt wrecked his car

**Question:** How was Matt's day?

**Answer:** Bad

# ChatGPT

Step 1

**Collect demonstration data and train a supervised policy.**

A prompt is sampled from our prompt dataset.



A labeler demonstrates the desired output behavior.



We give treats and punishments to teach...

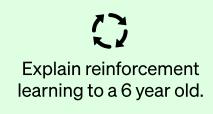
This data is used to fine-tune GPT-3.5 with supervised learning.



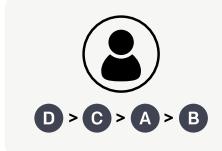
Step 2

**Collect comparison data and train a reward model.**

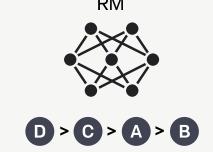
A prompt and several model outputs are sampled.



A labeler ranks the outputs from best to worst.



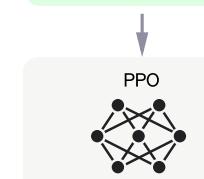
This data is used to train our reward model.



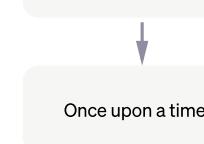
Step 3

**Optimize a policy against the reward model using the PPO reinforcement learning algorithm.**

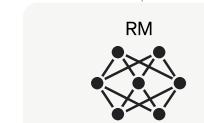
A new prompt is sampled from the dataset.



The PPO model is initialized from the supervised policy.



The policy generates an output.



The reward model calculates a reward for the output.



# LLMs

- How do they work → AKI
- How can I use them:
  - GPT (<https://platform.openai.com/docs/guides/gpt>), API Einstellung
  - Huggingface Framework
    - GPT
    - Llama
  - Private data
    - Embeddings
    - Vector DBs (storage & retrieval)
  - Agent Frameworks
    - LangChain (agents & tools)
    - Simulacra paper

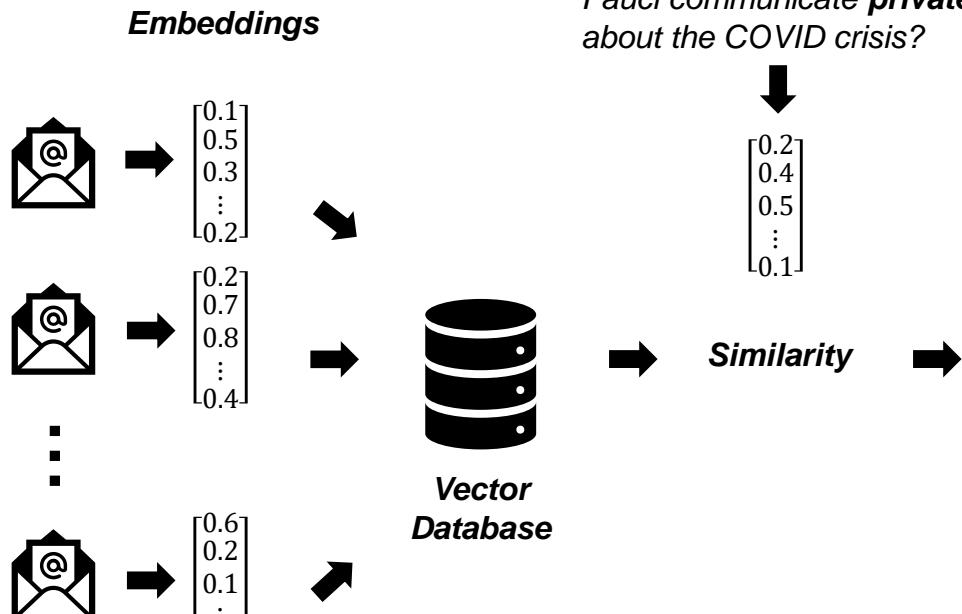
# Public vs. Private Data

*Prompt: How did Anthony Fauci communicate with **publicly** about the COVID crisis?*



*Overall, Dr. Fauci's communication strategy was multi-faceted and aimed at providing reliable, timely information to help guide the public through the uncertainties of the pandemic...*

*Query: How did Anthony Fauci communicate **privately** about the COVID crisis?*



*Prompt: How did Anthony Fauci communicate **privately** about the COVID crisis? Consider the following co...*

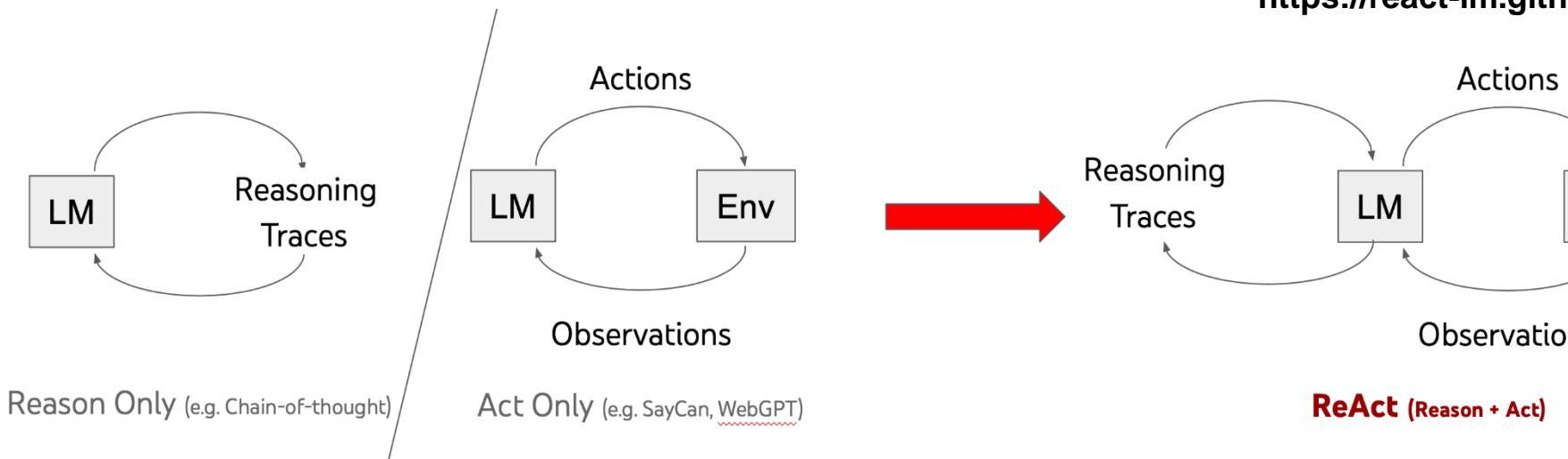
*"Gregg: I am surprised that you included me in your genuflect to no one but science and always, always, always my mind when it comes to public health. I have consistently corrected misstatements by others and continue to do so."*

*"Per email below, Mark Zuckerberg has extended his offers to do videos with you that we would be happy to seek clearance on for you to do, if you are amenable. These would have the weight and impact of television, really, moreso. Please advise if you want to do an interview, seek clearance with VP office and work with Patty to work out the logistics. But an even bigger deal is his offer to sconer we get that offer up the food-chain the better. Bill Halla heads-up about this. 'opportunity and he is standing by to discuss this with HES and WH com... didn't want him to do anything without you being a part of the offer. It OX if I hand this aspect off to Bill to determine who the best point of contact would be so the Administration can take advantage of this offer, so do you plan to call Mz? His cell number is in his message below. This is fucking crazy! Lmao"*

*"will write to or call Mark and tell him that am interested in this opportunity."*

# Agents Powered By LLMs

<https://react-lm.github.io>



**Question:** Aside from the Apple Remote, what other device can control the program Apple Remote was originally designed to interact with?

**Answer:** iPod

**Standard**



**Reason only**

**Thought:** Let's think step by step. Apple Remote was originally designed to interact with Apple TV. Apple TV can be controlled by iPhone, iPad, and iPod Touch. So the answer is iPhone, iPad, and iPod Touch.

**Answer:** iPhone, iPad, iPod Touch



**Act only**

**Act 1:** `Search[Apple Remote]`  
**Obs 1:** The Apple Remote is a remote control introduced...  
**Act 2:** `Search[Front Row]`  
**Obs 2:** Could not find [Front Row]. Similar: ...  
**Act 3:** `Search[Front Row (software)]`  
**Obs 3:** Front Row is a discontinued media center software ...

**ReAct**

**Thought 1:** I need to search Apple Remote and find the program it was originally designed to interact with.

**Act 1:** `Search[Apple Remote]`

**Obs 1:** The Apple Remote is a remote control introduced in October 2005 by Apple ... originally designed to control the Front Row media center program ...

**Thought 2:** Apple Remote was originally designed to control the Front Row media center program. I need to search for Front Row next and find what other device can control it.

**Act 2:** `Search[Front Row]`

**Obs 2:** Could not find [Front Row]. Similar: ['Front Row to Earth', 'Front Row Motorsports', 'Front Row (software)']

**Thought 3:** Front Row is not found. I need to search for Front Row (software) ...

**Act 3:** `Search[Front Row (software)]`

**Obs 3:** Front Row is a discontinued media center software ...

**Thought 4:** Front Row (software) is controlled by an Apple Remote or the keyboard function keys. So the answer is keyboard function keys.

# GPT4 as of Sep 11th 2023

**A**

Aside from the Apple Remote, what other device can control the program Apple Remote was originally designed to interact with?



The Apple Remote is primarily designed to interact with Apple devices like Apple TV, Mac computers, and some iPod docks. However, other devices and methods can also control these Apple products:



## Apple TV

1. **iPhone/iPad via Apple's Remote app:** You can control Apple TV using the Remote app available on iOS devices.
2. **Third-party remotes:** Universal remotes like Logitech Harmony can be programmed to work with Apple TV.
3. **Voice Control via Siri:** If you have a Siri-enabled remote or device, you can use voice commands to control the Apple TV.
4. **Game Controllers:** Some Bluetooth game controllers can be paired with Apple TV for control.
5. **HomeKit:** You can use HomeKit-compatible devices to create automation or control your Apple TV.

## Mac Computers

1. **Keyboard and Mouse:** The most direct way to control a Mac.
2. **iPhone/iPad via Sidecar or third-party apps:** Apps like Duet Display or software like TeamViewer can control a Mac.

# GPT4 as of Sep 11th 2023

**A**

Apple Remote was originally designed to interact with which program?



The Apple Remote was originally designed to interact with Front Row, a media center software application for Apple's Macintosh computers. Front Row provided a full-screen interface for accessing media stored on the computer, including music, photos, videos, and DVDs. The Apple Remote allowed users to easily navigate through Front Row's interface from a distance, making it convenient for use in a living room or home theater setting. Front Row was introduced in 2005 but was eventually discontinued as Apple shifted its focus towards other products and services.

**A**

Which other device than Apple Remote can control Front Row?



Front Row could also be controlled using the built-in keyboard and trackpad on a Mac, as well as through a compatible external keyboard or mouse. Additionally, various third-party remote controls that supported the infrared (IR) standard used by Apple could also be configured to work with Front Row. Universal remotes that could be programmed with the Apple Remote's IR codes could also be used.



For those who were more technically inclined, it was also possible to control Front Row using mobile apps that could send the appropriate commands over a local network, although this usually required some additional setup and software.

However, the Apple Remote was the device specifically designed by Apple for controlling Front Row and provided the most seamless user experience.

# Generative Agents



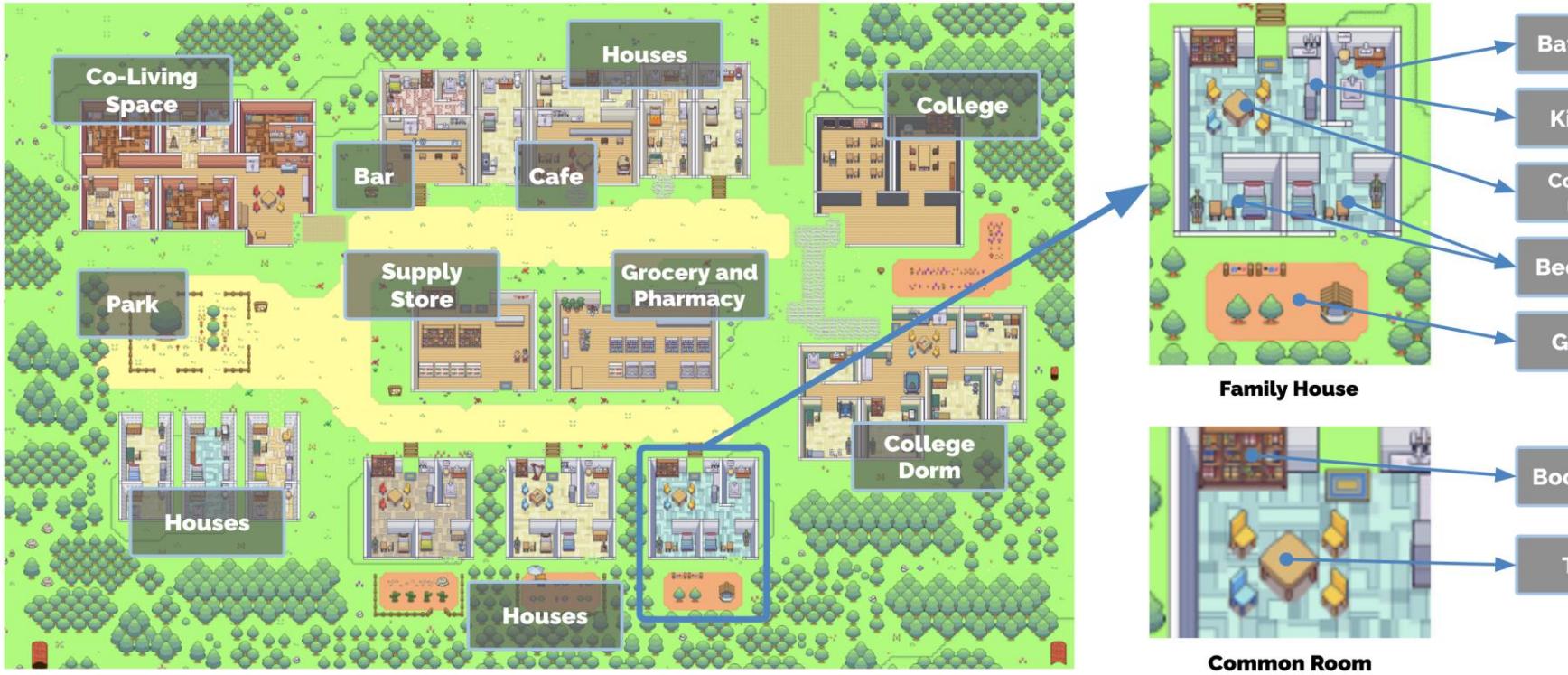
# Community of 25 unique agents inhabits Smallville

- Each agent is represented by a simple sprite avatar
- Authors provide one paragraph of natural language description to depict each agent's identity including
  - Occupation
  - Relationship with other agents
  - Seed memories
- Each semicolon-delimited phrase is entered into the agent's initial memory

John Lin is a pharmacy shopkeeper at the Willow Market and Pharmacy who loves to help people. He is always looking for ways to make the process of getting medication easier for his customers. John Lin is living with his wife, Mei Lin, who is a college professor, and son, Eddy Lin, who is a student studying music theory; John Lin loves his family very much; John Lin has known the couple next-door, Sam Moore and Jennifer Moreno, for a few years; John Lin thinks Sam Moore is a kind and nice man; John Lin knows his neighbor, Yuriko Yamamoto, well; John Lin knows of his neighbors, Tamara Taylor and Carmen Ortiz, but has not met them before; John Lin and Tom Moreno are colleagues at The Willows Market and Pharmacy. John Lin and Tom Moreno are friends and like to discuss local politics together; John Lin knows the Moreno family somewhat well – the husband Tom Moreno and the wife Jane Moreno.

# Sandbox Environment

- Server maintains a JSON data structure that contains information about each agent in the sandbox world, including
  - current location
  - description of their current action
  - sandbox object they are interacting with
- At each sandbox time step, the sandbox server parses the JSON for
  - any changes coming from the generative agents
  - moves the agents to their new positions
  - updates the status of any sandbox objects that the agents are interacting with (e.g., changing the status of the coffee machine from “idle” to “brewing coffee” if an agent’s action is “making espresso for a customer @ Hobbs Cafe: counter: coffee machine”).
- The sandbox server is also responsible for sending all agents and objects that are within a preset visual range for each agent to that agent’s memory, so the agent can react appropriately.
- The agent’s output action then updates the JSON, and the process loops for the next time step



**Figure 2: The Smallville sandbox world, with areas labeled. The root node describes the entire world, children describe subareas (e.g., houses, cafe, stores), and leaf nodes describe objects (e.g., table, bookshelf). Agents remember a subgraph that represents parts of the world they have seen, maintaining the state of those parts as they observed them.**

- Sandbox environment – areas and objects – are represented as tree data structure with an edge in the tree indicating a containment relationship in the sandbox world
- Tree is converted into natural language to ‘pass to generative agents: For instance, “stove” being a child of “kitchen” is rendered into “there is a stove in the kitchen.”
- Agents build individual tree representations of environment as they navigate it — subgraphs of the overall sandbox environment tree
- We initialize each agent with an environment tree capturing the spaces and objects that the agent should be aware of: the rooms and objects in their living quarters, their workplace, and commonly visited stores and landmarks
- As the agents navigate the sandbox world, they update this tree to reflect newly perceived areas
- Agents are not omniscient: their tree may get out of date as they leave an area, and is updated when they return

# Sandbox Environment

- To determine the appropriate location for each action, we traverse the agent's stored environment tree and flatten a portion of it into natural language to prompt the language model
- Recursively starting at the root of the agent's environment tree, we prompt the model to find the most suitable area
- For example, if Eddy's agent indicated that he should *take a short walk around his workspace*:

*[Agent's Summary Description] Eddy Lin is currently in The Lin family's house: Eddy Lin's bedroom: desk) that has Mei and John Lin's bedroom, Eddy Lin's bedroom, common room, kitchen, bathroom, and garden.*

*Eddy Lin knows of the following areas: The Lin family's house, Johnson Park, Harvey Oak Supply Store, The Willows Market and Pharmacy, Hobbs Cafe, The Rose and Crown Pub.*

*\* Prefer to stay in the current area if the activity can be done there. Eddy Lin is planning to take a short walk around his workspace. Which area should Eddy Lin go to?*

- This outputs *The Lin family's house*
- We then use the same process recursively to determine the most appropriate subarea within the chosen area until we reach a leaf node of the agent's environment tree
- In the example above, the result of this traversal is *The Lin family's house: garden: house garden*
- Finally, we use traditional game path algorithms to animate the agent's movement so that it travels to the location indicated by the leaf node

# Sandbox Environment

- When an agent executes an action on an object, we prompt the language model to ask what happens to the state of the object. For example, if Isabella's generative agent outputs the action "making espresso for a customer", a query to the language model indicates a response that the state of the coffee machine in Hobbs Cafe should change from "off" to "brewing coffee".

## Prompt Template

*generate\_obj\_event\_v1.txt*

*Variables:*

*!<INPUT 0>! -- Object name  
!<INPUT 1>! -- Persona name  
!<INPUT 2>! -- Persona action event description  
!<INPUT 3>! -- Object name  
!<INPUT 4>! -- Object name*

*<commentblockmarker>###</commentblockmarker>*

*Task: We want to understand the state of an object that is being used by someone.*

*Let's think step by step.*

*We want to know about !<INPUT 0>!'s state.*

*Step 1. !<INPUT 1>! is at/using the !<INPUT 2>!.*

*Step 2. Describe the !<INPUT 3>!'s state:*

## Prompt Instance

*Task: We want to understand the state of an object that is being used by someone.*

*Let's think step by step.*

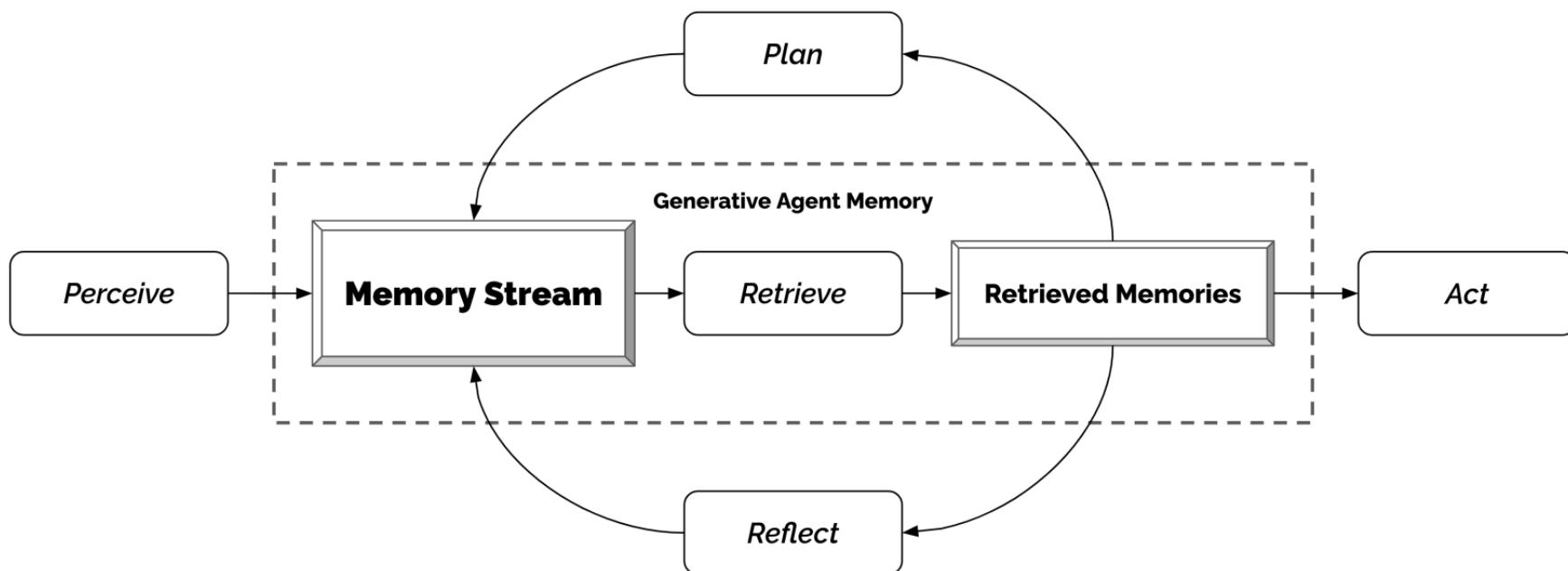
*We want to know about coffee machine's state.*

*Step 1. Isabella is at/using the making espresso for a customer.*

*Step 2. Describe the coffee machine's state:*

# Generative Agents Architecture

- Agents perceive their environment
- Perceptions are saved in agent's memory stream
- Based on perceptions, relevant memories are retrieved and used to determine an action
- Retrieved memories are also used to form longer-term plans and create higher-level reflections, both of which are entered into the memory stream for future use



# Memory and Retrieval

## Memory

- Most basic element of the memory stream is an observation, which is an event directly perceived by an agent (e.g., events on tiles within a visible region)
- Common observations include behaviors performed by the agent themselves or behaviors that agents perceive being performed by other agents or non-agent objects

## Retrieval

- **Recency:** Assigns higher score to memory objects that were recently accessed, so that events from a moment ago or this morning are likely to remain in the agent's attentional sphere. Here recency is treated as exponential decay function over the number of sandbox game hours since the memory was last retrieved (decay factor is 0.995)
- **Importance:** Determined by LLM (see prompt below)
- **Relevance:** Cosine similarity with query

### Prompt to determine importance

On the scale of 1 to 10, where 1 is purely mundane (e.g., brushing teeth, making bed) and 10 is extremely poignant (e.g., a break up, college acceptance), rate the likely poignancy of the following piece of memory.

Memory: buying groceries at The Willows Market and Pharmacy

### Memory Stream

2023-02-13 22:48:20: desk is idle  
2023-02-13 22:48:20: bed is idle  
2023-02-13 22:48:10: closet is idle  
2023-02-13 22:48:10: refrigerator is idle  
2023-02-13 22:48:10: Isabella Rodriguez is stretching  
2023-02-13 22:33:30: shelf is idle  
2023-02-13 22:33:30: desk is neat and organized  
2023-02-13 22:33:10: Isabella Rodriguez is writing in her journal  
2023-02-13 22:18:10: desk is idle  
2023-02-13 22:18:10: Isabella Rodriguez is taking a break  
2023-02-13 21:49:00: bed is idle  
2023-02-13 21:48:50: Isabella Rodriguez is cleaning up the kitchen  
2023-02-13 21:48:50: refrigerator is idle  
2023-02-13 21:48:50: bed is being used  
2023-02-13 21:48:10: shelf is idle  
2023-02-13 21:48:10: Isabella Rodriguez is watching a movie  
2023-02-13 21:19:10: shelf is organized and tidy  
2023-02-13 21:18:10: desk is idle  
2023-02-13 21:18:10: Isabella Rodriguez is reading a book  
2023-02-13 21:03:40: bed is idle  
2023-02-13 21:03:30: refrigerator is idle  
2023-02-13 21:03:30: desk is in use with a laptop and some papers on it

...

**Q. What are you looking forward to the most right now?**

Isabella Rodriguez is excited to be planning Valentine's Day party at Hobbs Cafe on February 14th from 5pm and is eager to invite everyone to attend the party.

retrieval	=	recency	importance	relevance
2.34	=	0.91	+ 0.63	+ 0.85

ordering decorations for the party

2.21	=	0.87	+ 0.63	+ 0.71
------	---	------	--------	--------

researching ideas for the party

2.20	=	0.85	+ 0.73	+ 0.68
------	---	------	--------	--------

...



I'm looking forward to the Valentine's Day party that I'm planning at Hobbs Cafe!

# Reflections: Generalize And Make Inferences

- Generative agents, when equipped with only raw observational memory, struggle to generalize or make inferences.
- Consider a scenario in which Klaus Mueller is asked by the user: "If you had to choose one person of those you know to spend an hour with, who would it be?"
- With access to only observational memory, the agent simply chooses the person with whom Klaus has had the most frequent interactions: Wolfgang, his college dorm neighbor. Unfortunately, Wolfgang and Klaus only ever see each other in passing, and do not have deep interactions.
- A more desirable response requires that the agent generalize from memories of Klaus spending hours on a research project to generate a higher level reflection that Klaus is passionate about research, and likewise recognize Maria's passion for putting in effort into her own research (albeit in a different field), enabling a reflection that they share a common interest.
- With the approach below, when Klaus is asked who to spend time with, Klaus chooses Maria instead of Wolfgang.

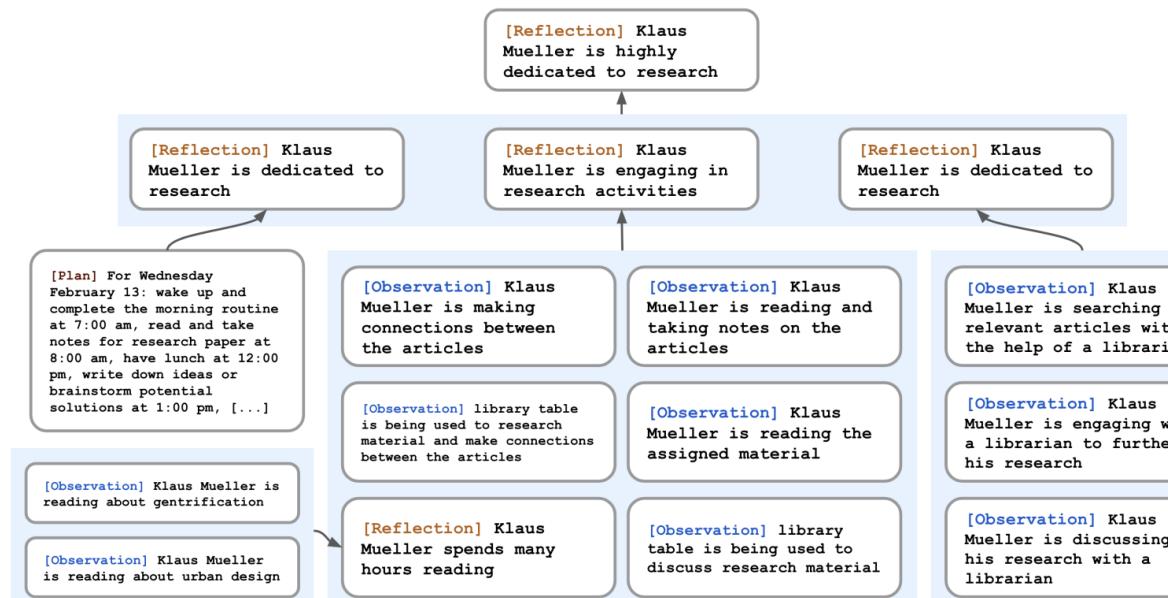


Figure 7: A reflection tree for Klaus Mueller. The agent's observations of the world, represented in the leaf nodes, are synthesized to derive Klaus's self-notion that he is highly dedicated to his research.

# Reflection: How To Decide What To Generalize And Infer?

## 3. Step: Prompt to determine what to reflect on

"Klaus Mueller is reading a book on gentrification", "Klaus Mueller is conversing with a librarian about his research project", "desk at the library is currently unoccupied", ...

Given only the information above, what are 3 most salient high-level questions we can answer about the subjects in the statements?

1. What topic is Klaus Mueller passionate about?
2. What is the relationship between Klaus Mueller and Maria Lopez?
3. ...

## 2. Step: Get 100 most recent records in agent's memory stream

**1. Step: Initiate reflection whenever sum of importance scores for latest events perceived by the agent exceeds a threshold (e.g. 150)**

**Memory Stream**

## 4. Step: Use query to retrieve relevant memories (incl. reflections)

Query: What topic is Klaus Mueller is passionate about?

## 5. Step: Prompt to generate reflections

Statements about Klaus Mueller

1. Klaus Mueller is writing a research paper
2. Klaus Mueller enjoys reading a book on gentrification
3. Klaus Mueller is conversing with Ayesha Khan about exercising [...]

What 5 high-level insights you infer from the above statements? (example format: insight (because of 1, 5, 3))

## 6. Step: Update memory stream

1. Klaus Mueller is dedicating his research on gentrification (because of 1, 2, 8, 15, 2, ...)

# Planning

## 1. Step: Prompt to create plan that outlines day's agenda in broad strokes

Name: Eddy Lin (age: 19)  
Innate traits: friendly, outgoing, hospitable  
Eddy Lin is a student at Oak Hill College studying music theory and composition. He loves to explore different musical styles and is always looking for ways to expand his knowledge. Eddy Lin is working on a composition project for his college class. He is taking classes to learn more about music theory. Eddy Lin is excited about the new composition he is working on but he wants to dedicate more hours in the day to work on it in the coming days  
On Tuesday February 12, Eddy 1) woke up and completed the morning routine at 7:00 am, [ . . . ] 6) got ready to sleep around 10 pm.  
Today is Wednesday February 13. Here is Eddy's plan today in broad strokes: 1)



- 1) Wake up and complete the morning routine at 8:00 am
- 2) Go to Oak Hill College to take classes starting 10:00 am
- [ . . . ]
- 5) Work on his new music composition from 1:00 pm to 5:00 pm
- 6) Have dinner at 5:30 pm
- 7) Finish school assignments and go to bed by

[Agents summary description]

[Summary of agent's previous day]

Memory Stream

## 5) Work on his new music composition from 1:00 pm to 5:00 pm

## 2. Step: Plan his day by decomposing it into hourly segments

- 1:00 pm: start by brainstorming some ideas for his music composition
- [...]
- 4:00 pm: take a quick break to recharge his creative energy before reviewing and polishing his composition.

## 3. Step: Plan his day by decomposing it into mini. scheduled tasks

- 4:00 pm: grab a light snack as a piece of fruit, a granola bar or some nuts
- 4:05 pm: take a short walk around his workspace
- [...]
- 4:50 pm: take a few minutes to clean up his workspace

# Use Prompt Templates With Examples

```
Variables:  
!<INPUT 0>! -- Commonset  
!<INPUT 1>! -- Surrounding schedule description  
!<INPUT 2>! -- Persona first name  
!<INPUT 3>! -- Persona first name  
!<INPUT 4>! -- Current action  
!<INPUT 5>! -- curr time range  
!<INPUT 6>! -- Current action duration in min  
!<INPUT 7>! -- Persona first names  
  
<commentblockmarker>###</commentblockmarker>  
Describe subtasks in 5 min increments.  
---  
Name: Kelly Bronson  
Age: 35  
Backstory: Kelly always wanted to be a teacher, and now she teaches kindergarten. During the week, she dedicates herself to her students, but  
Personality: sweet, gentle, meticulous  
Location: Kelly is in an older condo that has the following areas: {kitchen, bedroom, dining, porch, office, bathroom, living room, hallway}  
Currently: Kelly is a teacher during the school year. She teaches at the school but works on lesson plans at home. She is currently living a  
Daily plan requirement: Kelly is planning to teach during the morning and work from home in the afternoon.s  
  
Today is Saturday May 10. From 08:00am ~09:00am, Kelly is planning on having breakfast, from 09:00am ~ 12:00pm, Kelly is planning on working  
In 5 min increments, list the subtasks Kelly does when Kelly is working on the next day's kindergarten lesson plan from 09:00am ~ 12:00pm (t  
1) Kelly is reviewing the kindergarten curriculum standards. (duration in minutes: 15, minutes left: 165)  
2) Kelly is brainstorming ideas for the lesson. (duration in minutes: 30, minutes left: 135)  
3) Kelly is creating the lesson plan. (duration in minutes: 30, minutes left: 105)  
4) Kelly is creating materials for the lesson. (duration in minutes: 30, minutes left: 75)  
5) Kelly is taking a break. (duration in minutes: 15, minutes left: 60)  
6) Kelly is reviewing the lesson plan. (duration in minutes: 30, minutes left: 30)  
7) Kelly is making final changes to the lesson plan. (duration in minutes: 15, minutes left: 15)  
8) Kelly is printing the lesson plan. (duration in minutes: 10, minutes left: 5)  
9) Kelly is putting the lesson plan in her bag. (duration in minutes: 5, minutes left: 0)  
---  
!<INPUT 0>!  
!<INPUT 1>!  
In 5 min increments, list the subtasks !<INPUT 2>! does when !<INPUT 3>! is !<INPUT 4>! from !<INPUT 5>! (total duration in minutes !<INPUT 6>)
```

# Further Topics

- Reaction
- Dialogue
- Updating agents summary descriptions (how is agent changing)
- [https://github.com/joonspk-research/generative\\_agents](https://github.com/joonspk-research/generative_agents)
- <https://arxiv.org/pdf/2308.11432.pdf>



# Project 1: Simulate Social Platform With Generative Agents

- Provide initial description of agent (possibly hobbies, interests, CV, goals, etc.)
- For each member, decide to have conversation (on a daily basis)
- Feed conversation into memory stream
- Generate summary and feed into memory stream
- Rate each entry how interesting it was
- Generate higher-level reflections per chat partner and across chat partners
- Retrieve relevant content for conversations (history for particular member, higher-level reflections from other conversations, etc.)
- Allow user feedback/input
- Goal could be to generate as deep conversations as possible (evaluation via LLM)

# Project 2: Digital Twin

- Build a digital version of yourself
- Connectors to email, X, LinkedIn, Moodle, Whatsapp, Instagram, FB etc.
- Personal files
- Microphone transcript (record meetings, Zoom, etc.)
- Meta Smart Glasses (record images, videos)
- Let LLM extract additional features (e.g. personality traits, high-level reflections)
- Also check Quivr (<https://www.quivr.app/>), Rewind (<https://www.rewind.ai/>), Augment (<https://www.augment.co/>)
- Local version with ChromaDB
- UI to navigate personal data (<https://github.com/facebookresearch/personal-timeline/tree/main>)



## Query your personal timeline

Enter a question or "clear" to clear all commands.

ChatGPT  Retrieval-based  View-based

Show me some photos of plants in my neighborhood

Which cities did I visit when I traveled to Japan? How many books did I purchase in April?

Welcome to TL-QA

TL-QA \$ Show me some photos of plants in my neighborhood

Retrieval-based: The photos of plants in my neighborhood are Sweet pepperbush, Fernleaf lavender, Lupinus mutabilis.

TL-QA \$ Which cities did I visit when I traveled to Japan?

Retrieval-based: I visited Tokyo (Minato, Bunkyo, Meguro, Sanmu, Chiba Prefecture), Roppongi Hills, Tokyo Prince Hotel, Asahi Shokudo, University of Tokyo (Hongo Campus), 中里橋, Roppongi Hills Mori Tower, 191L, Grand Hyatt Tokyo, Naka-Meguro, and Narita in Japan.

TL-QA \$ How many books did I purchase in April?

Retrieval-based: I purchased 8 books in April.

TL-QA \$

Query result    **Retrieval Results**    Map    Details

April

Wed Apr 03 2019

The Boys

Wed Apr 03 2019 20:41:00 GMT-0700 (Pacific Daylight Time)



More details

Fri Apr 05 2019

# Project 3: Prompt Optimizer

- Generalization of Seller-Buyer problem
- Given a prompt and an output, let LLM rate the output
- Continue to let LLM reformulate prompt, generate output, and rate output
- Based on above triples, let LLM reformulate an ideal prompt
- Check out the following projects:
  - <https://github.com/thunlp/ChatEval>
  - <https://arxiv.org/abs/2305.10142>



Large Language Model (LLM) based agent

Single-agent method

Question: How can I improve my time management skills?

ASSISTANT1: Improving your time management skills involves ...

ASSISTANT2: Here are some tips to improve your time management, like ...



After carefully reviewing the responses of both response ... I think ASSISTANT1 is better.

Multi-agent debat



After discussing thoroughly with my co-workers, we are convinced that ASSISTANT2 is better based on the reason ...

# Project 4: Simulate/Moderate Multi-Person Discussions

- Description of meeting objectives
- Description of roles (incl. moderator)
- Simulate debates (with interruption by feeding input token by token?)
- How to add depth to the discussion?
  - Preparation of arguments and viewpoints
  - Generate complex argument maps (e.g. argdown.org) as input for agents
- Let LLM rate meeting and each participant (include micro-feedback:  
<https://www.youtube.com/watch?v=tUu-6UxuGr8>)

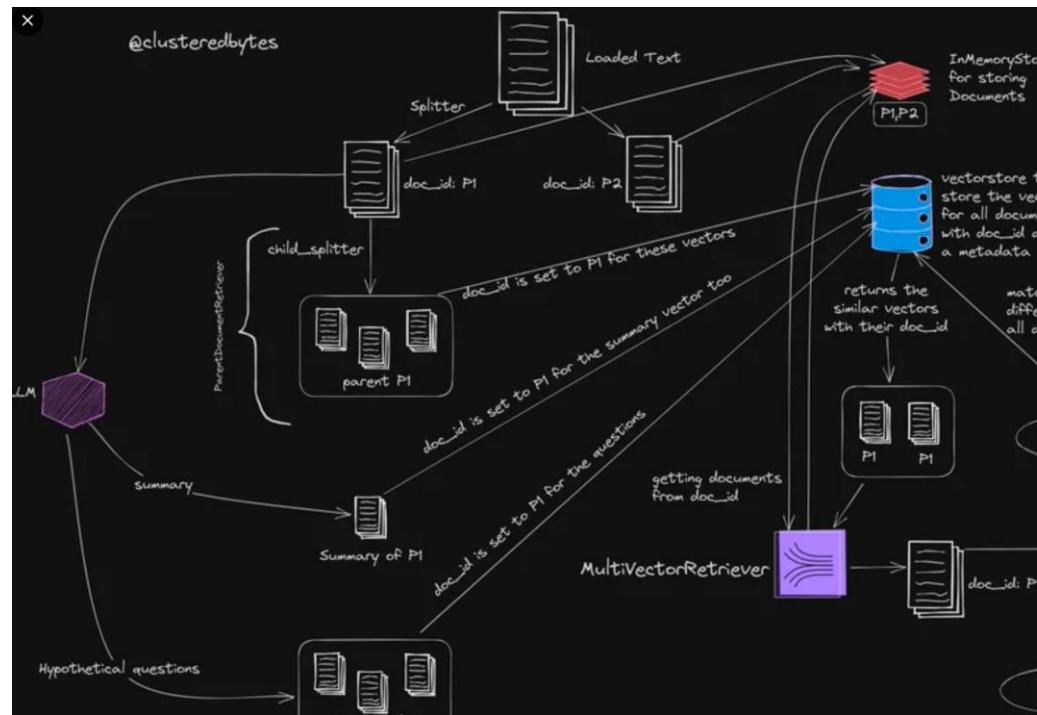


# Project 5: Recursively Breaking Down And Executing Complex Tasks Using Tools

- Example: writing a book
- Use LLM to generate list of high-level book characteristics (topic, genre, length, main characters, attributes such as funny, plot twist, sad etc.)
- Give detailed description for some characteristics or let LLM choose
- LLM generate synopsis
- LLM breaks down synopsis into chapters
- LLM breaks down chapters into paragraphs
- Let LLM take role of book critic to generate feedback per paragraph, chapter, synopsis
- Let LLM rework pieces based on feedback
- Use search API to do research and enrich paragraphs, chapter, synopsis
- How to ensure that everything fits together?
  - Summarization
  - Retrieval
- Other areas:
  - Games
  - Theses

# Project 6: Exploring Vector DBs

- Imagine digital twin with plenty of information in vector DB
- Documents (conversation, email, message etc.) is typically split into smaller chunks in order to make only relevant parts can be retrieved
- Chunks might have too much or too less information for a given query
- Strategies:
  - Add document summary
  - Add hypothetical questions
  - Based on query, extract large  $k$  and perform recursive clustering to provide different notions of results
  - Enrich DB with high-level reflections from clusters of topics (e.g. psychograms for individuals)
  - Visual exploration of vector DB (<https://projector.tensorflow.org/>)



# Project 7: Develop Streamlit App To Showcase AI Driven Solutions

- Example Learning Platform (e.g. Moodle)
- Think about compelling use cases
- Use Moodle API to get and post data:  
[https://docs.moodle.org/dev/Web service API functions](https://docs.moodle.org/dev/Web_service_API_functions)
- Use chat interface to communicate with Moodle and use LLM to implement use cases
- Student: „Please submit document XY for today's assignment and provide some initial feedback“
- Teacher: „Please extract all submissions from Student XY, summarize and give first impression on strengths and weaknesses.“
- Streamlit: <https://streamlit.io/gallery?category=llms>

# Project 8: Generative Agents For Developing Software

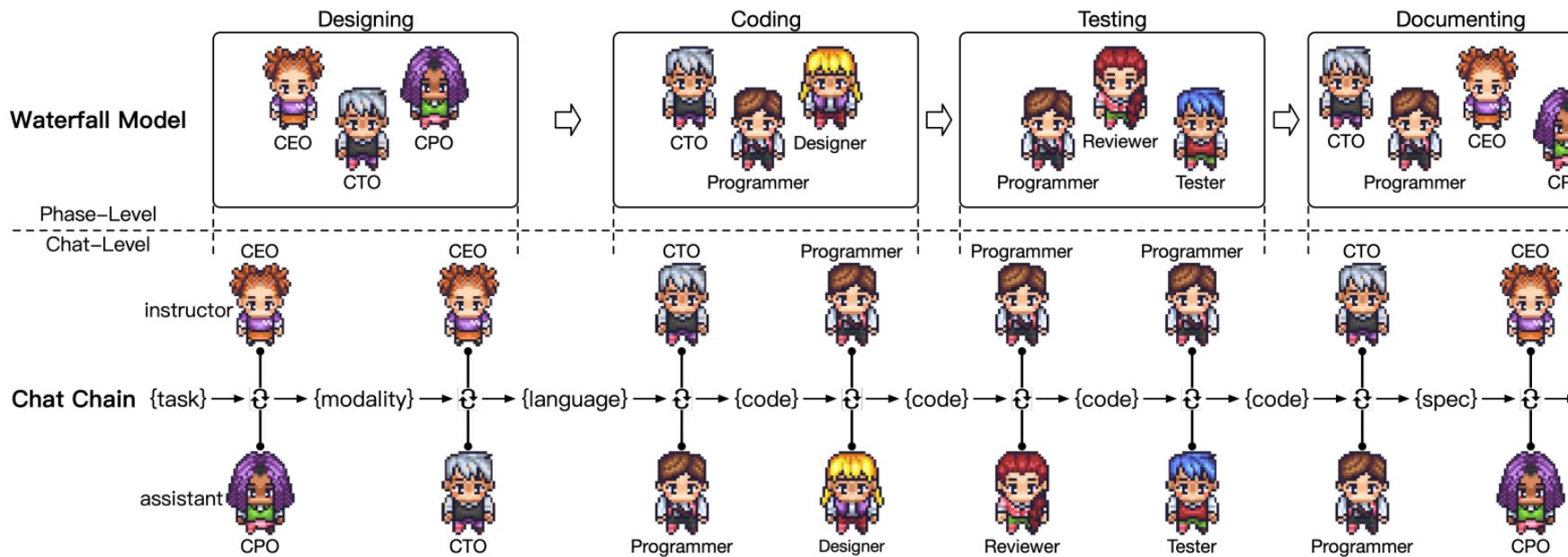


Figure 2: The proposed architecture of CHATDEV consists of phase-level and chat-level components. At the phase level, the waterfall model is used to break down the software development process into four sequential phases. At the chat level, each phase is further divided into atomic chats. These chats involve task-oriented role-playing between two agents, promoting collaborative communication. The communication follows an instruction-following style, where agents interact to accomplish specific subtask within each chat.