MidTerm Review

DATA/COMPSCI 182 Deep Learning
Lecture 15 10/17/2024

Bias, Variance, Training, Learning Rate

Which of the following is the best description of the bias-variance tradeoff in machine learning?

A. Bias refers to the model's ability to generalize to new data, while variance refers to the model's error on the training data.

B. High bias leads to overfitting, while high variance leads to underfitting.

C. Bias is the error introduced by approximating a real-world problem, while variance is the model's sensitivity to the specific training data used.

D. Bias and variance are unrelated concepts in deep learning.

What could happen if the learning rate is set too high during the training of a neural network?

A. The model will converge too slowly and may not reach the optimal solution.

B. The model will converge to a local minimum but may take a long time to do so.

C. The model may overshoot the optimal solution, resulting in unstable training.

D. The model will train successfully but may have higher bias.

Bias, Variance, Training, Learning Rate

Which of the following is true about training with a small dataset?

A. A small dataset always leads to high variance in the model's predictions.

B. A small dataset can cause underfitting as the model fails to learn complex patterns.

C. A small dataset may result in overfitting, as the model can memorize the data rather than generalize.

D. Training with a small dataset results in lower training time but improves test accuracy.

Which of the following adjustments can help reduce overfitting in a deep neural network?

A. Increasing the number of epochs.

B. Decreasing the size of the training data.

C. Increasing the size of the model (more layers and parameters).

D. Applying regularization techniques like dropout or L2 regularization.

If a neural network is underfitting the training data, which of the following actions is most likely to help?

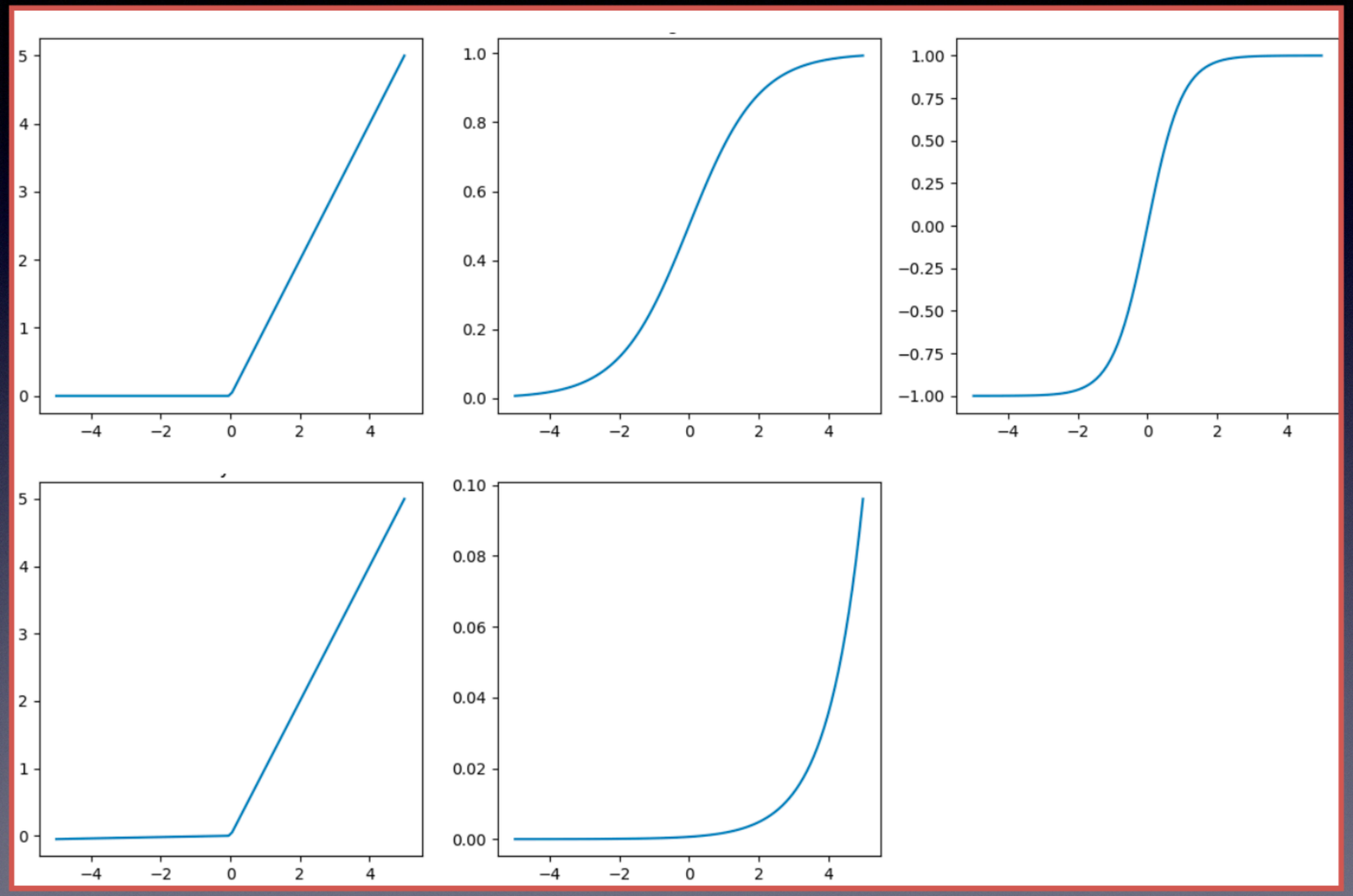
A. Decreasing the learning rate.

B. Increasing the number of layers or neurons in the model.

C. Decreasing the size of the training dataset.

D. Adding more regularization to the model.

Gradient Issues



Gradients

$$y = Wx + b$$

$$\nabla_x L = \frac{\partial L}{\partial x} = \frac{\partial y}{\partial x} \cdot \frac{\partial L}{\partial y}$$

$$\frac{\partial y}{\partial x} = W$$

$$\nabla_x L = W^T \cdot \frac{\partial L}{\partial y}$$

$$\nabla_W L = \frac{\partial y}{\partial W} \cdot \frac{\partial L}{\partial y}$$

$$\frac{\partial y}{\partial W} = x^T$$

$$\nabla_W L = \frac{\partial L}{\partial y} \cdot x^T$$

$$\nabla_b L = \frac{\partial L}{\partial y}$$

Parameter Initialization

$$a_i^{(l)} = \sum_{j=1}^{M} w_{ij} z_j^{(l-1)}$$
$$z_i^{(l)} = \mathrm{ReLU}(a_i^{(l)})$$

$$\mathbb{E}[a_i^{(l)}] = 0$$
$$\mathrm{var}[z_j^{(l)}] = \frac{M}{2}\epsilon^2\lambda^2$$

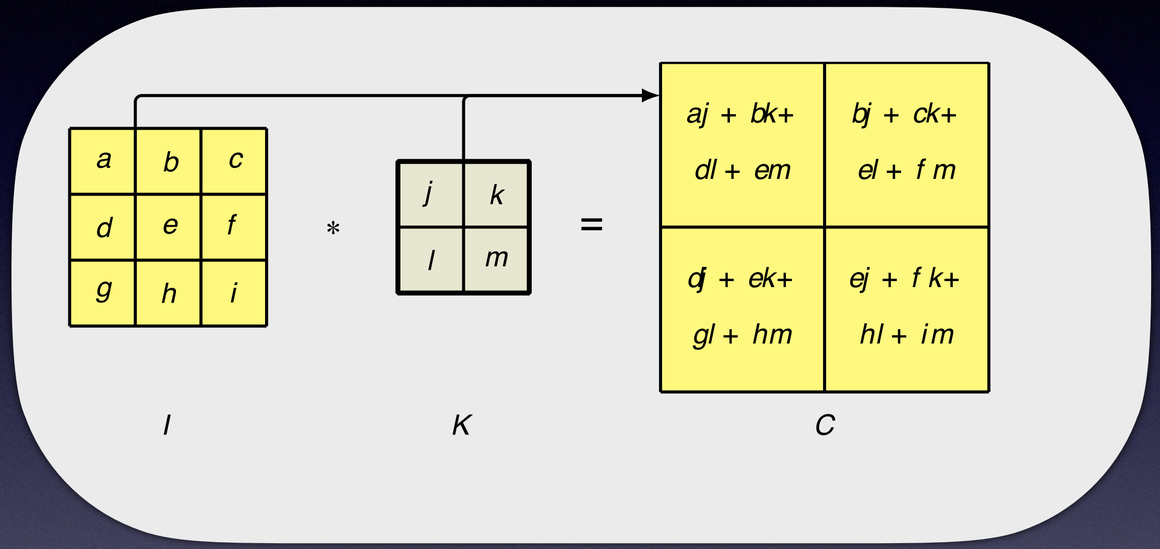
M is the number of connections that send input to unit I

- Bishop Book 7.2.5
- *Symmetry breaking* ?
- “He initialization” **(Gaussian)**

Learning Rate

CNNs

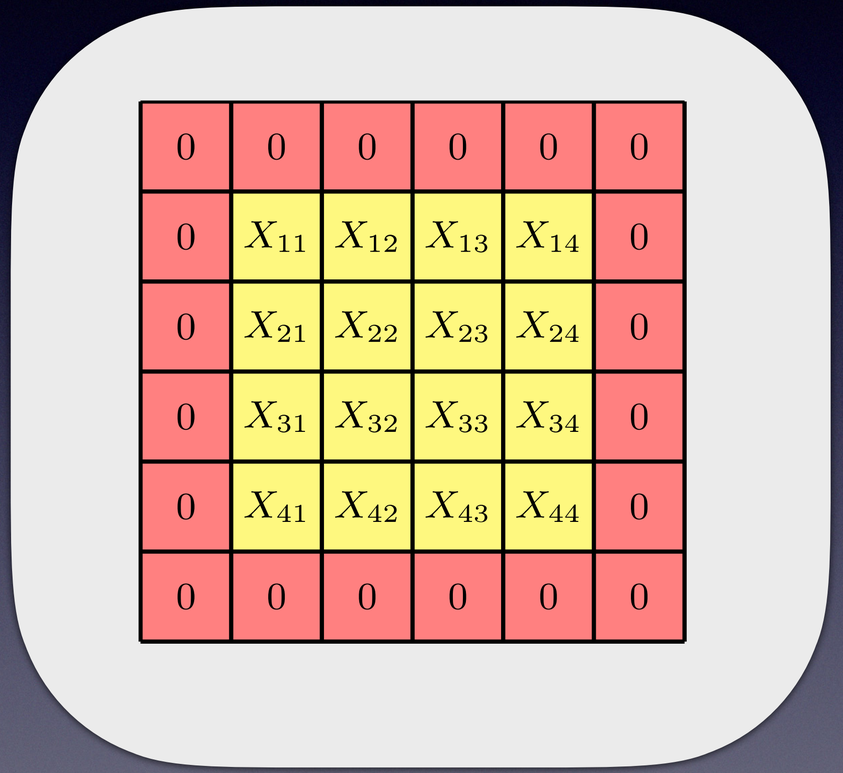
Convolution



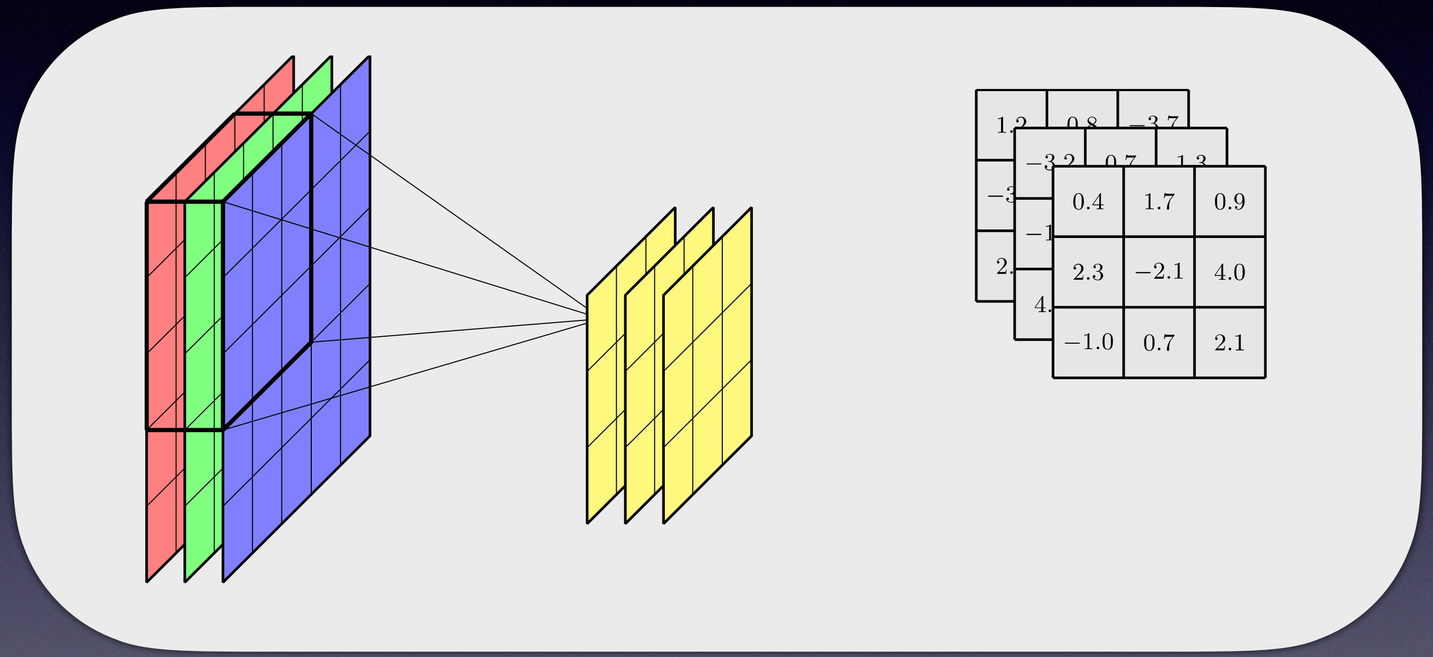
-1	-1	-1
0	0	0
1	1	1

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

Padding, Strides

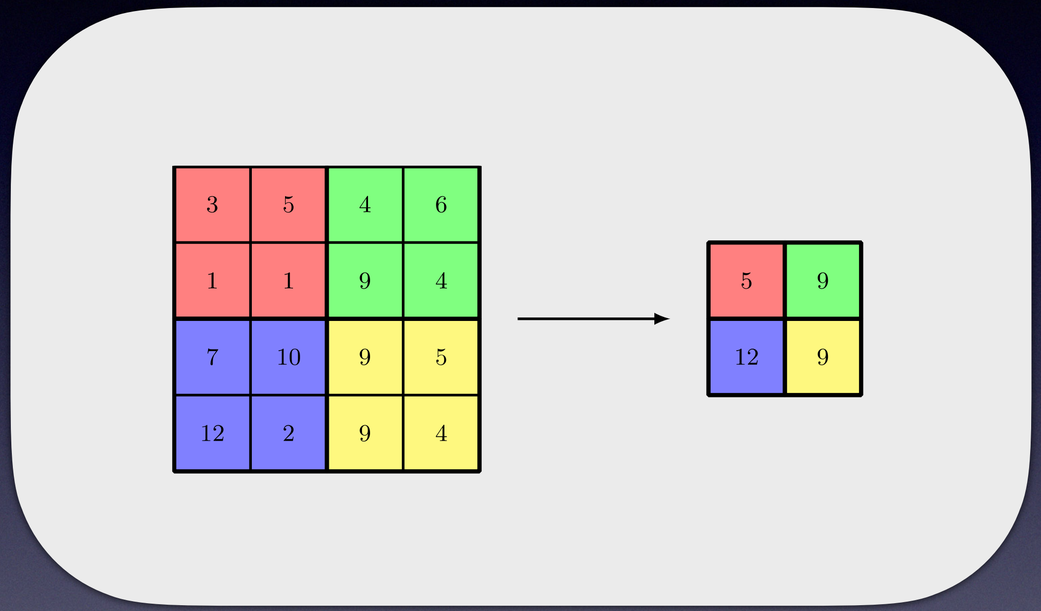


Multi-dimensional Convolutions



- Channels
- Filter Tensor dimensionality: $M \times M \times C \times C_{OUT}$
- Number of parameters: $(M^2C + 1)\, C_{OUT}$

Pooling



Convolution Exercise

1	5	2	0	1
0	3	1	2	4
4	0	1	1	3
2	1	0	4	1
0	1	2	1	2

1	0	1
0	1	0
1	0	1

?	?
?	?

- no padding, but stride of 2

Edge Detection

0	50	50	0	0	0
0	50	50	0	0	0
0	50	50	0	0	0
0	50	50	0	0	0
0	50	50	0	0	0

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

150	-150	-150	0
150	-150	-150	0
150	-150	-150	0

(a)

(b)

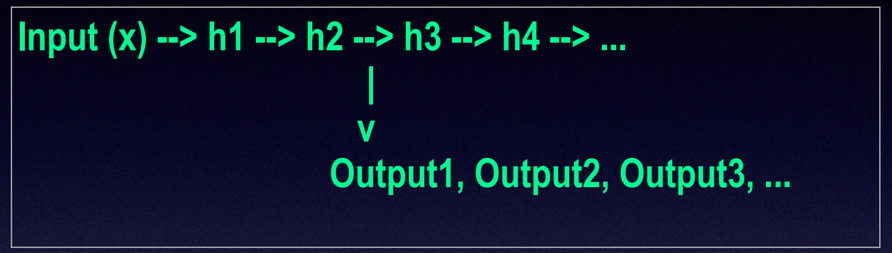
(c)

RNNs

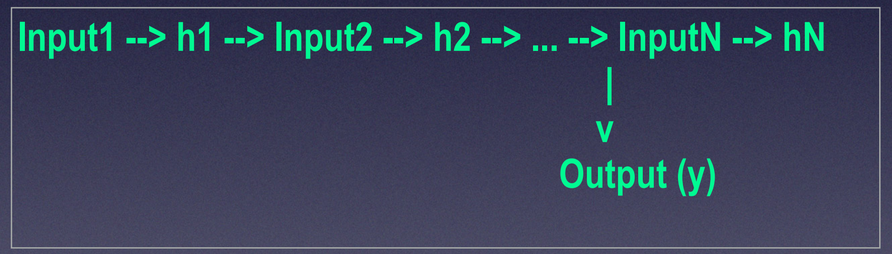
- one-to-many
- many-to-one
- many-to-many

RNN

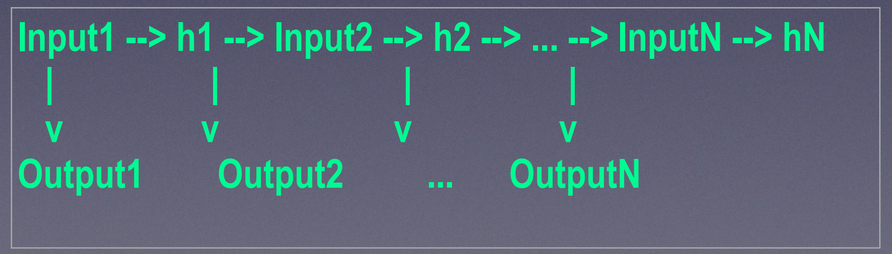
- one-to-many RNN: a **single input** is provided at the first time step, and the network produces a **sequence of outputs** over multiple time stepsIn a one-to-many RNN, a **single input** is provided at the first time step, and the network produces a **sequence of outputs** over multiple time steps



- many-to-one RNN: a **sequence of inputs** is processed, and a **single output** is produced after reading the entire sequence



- many-to-many RNN: a **sequence of inputs** is processed, and a **sequence of outputs** is produced, where the output at each time step may depend on the input sequence and previous outputs.

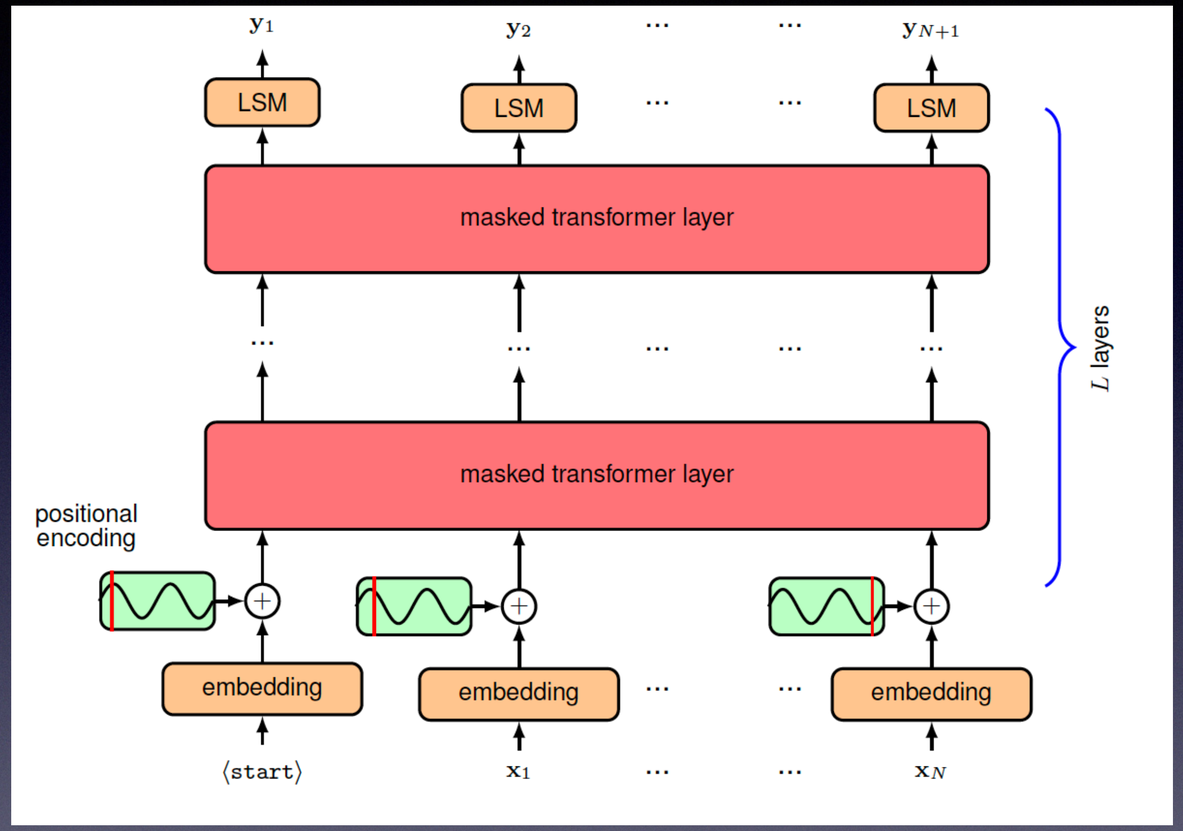


RNN operation

$$h_t = \tanh(W_{hh}h_{t-1} + W_{xh}x_t + b_h)$$

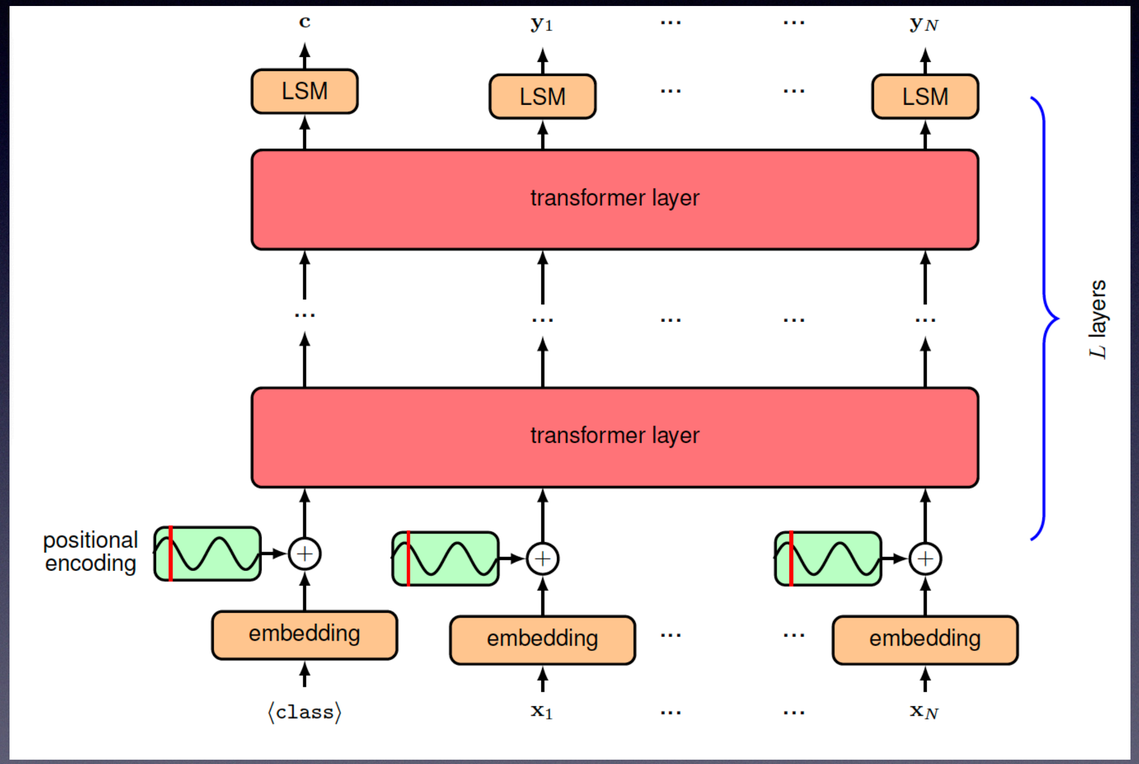
$$y_t = W_{hy}h_t + b_y$$

Transformers



- Parallel processing of multiple sentences
  - Length of sequences ?

Transformers



Cheat sheet : unsolicited advice

Make one ! The very process is one of the best ways of reviewing your material