EEP 596: Adv Intro ML || Lecture 15

Dr. Karthik Mohan

Univ. of Washington, Seattle

February 23, 2023

Last Time

a. Anomaly Detection
b. Deep Learning Basics

Today

- Deep Learning Fundamentals
- Auto Encoders
- Deep Learning in NLP
- Sequence to Sequence models

Tensorflow Playground Demo

Walk through

Tensorflow Playground Demo

Hyper-parameters in Deep Learning

ICE #1: Which of the following is not a hyper-parameter in deep learning?

1. Learning rate
2. Number of Hidden Layers
3. Number of neurons per hidden layer
4. All of the above

Hyper-parameters in Deep Learning

Hyper-parameters

1. Learning rate → Practically not a hyper param - choose a LR scheduler
2. Number of Hidden Layers
3. Number of neurons per hidden layer

Hyper-parameters in Deep Learning

Hyper-parameters

1. Learning rate
2. Number of Hidden Layers
3. Number of neurons per hidden layer
4. Type of non-linear activation function used

Hyper-parameters in Deep Learning

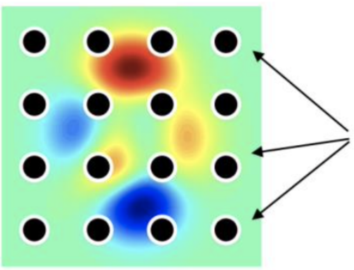
Hyper-parameters

1. Learning rate
2. Number of Hidden Layers
3. Number of neurons per hidden layer
4. Type of non-linear activation function used
5. Anything else?] → other hyper-params depending on architecture?

(e.g. Conv. stride length in CNNs)

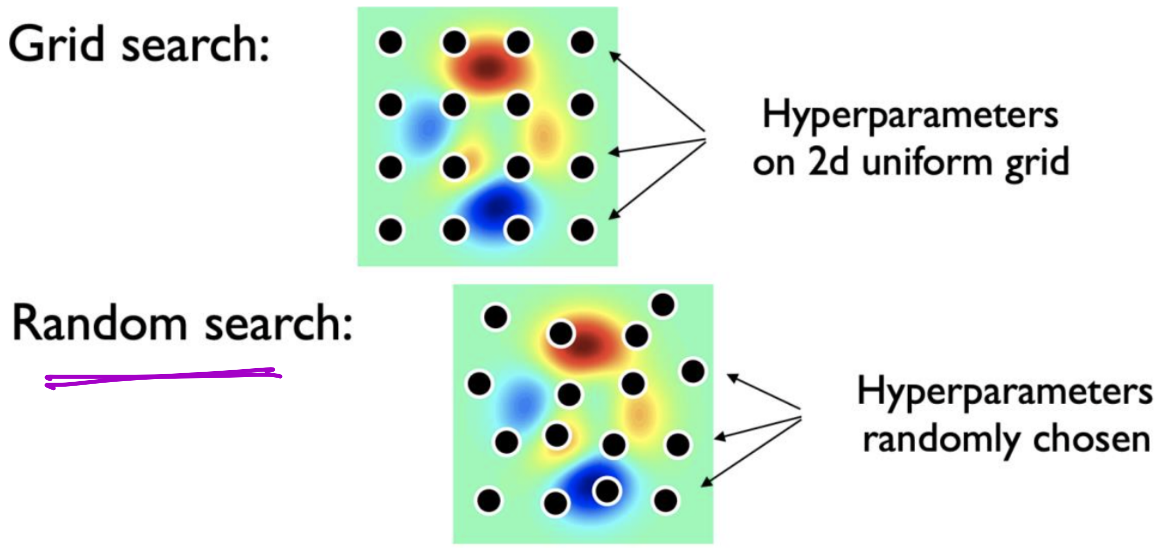
Hyper-parameter tuning methods

Grid search:

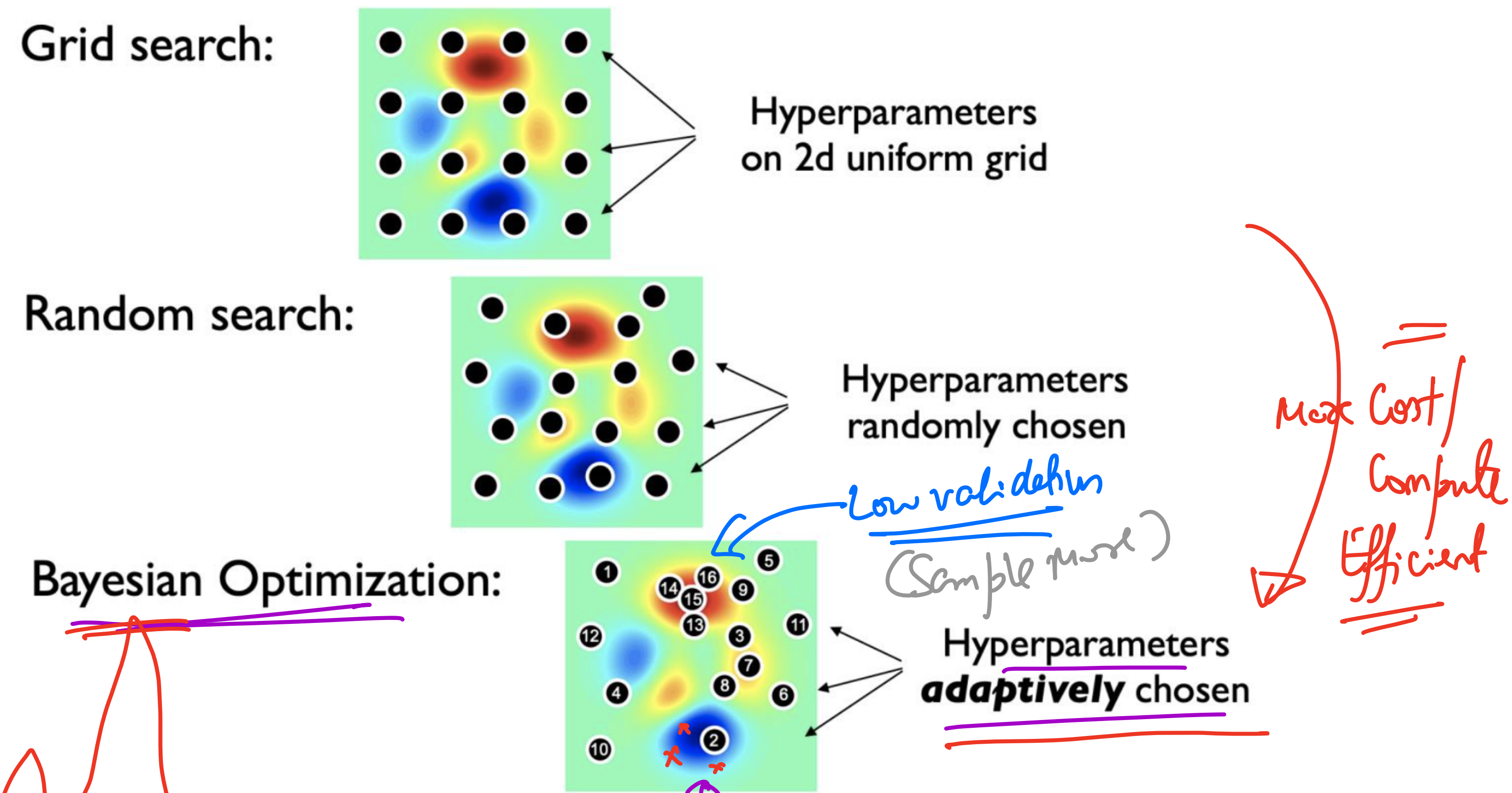


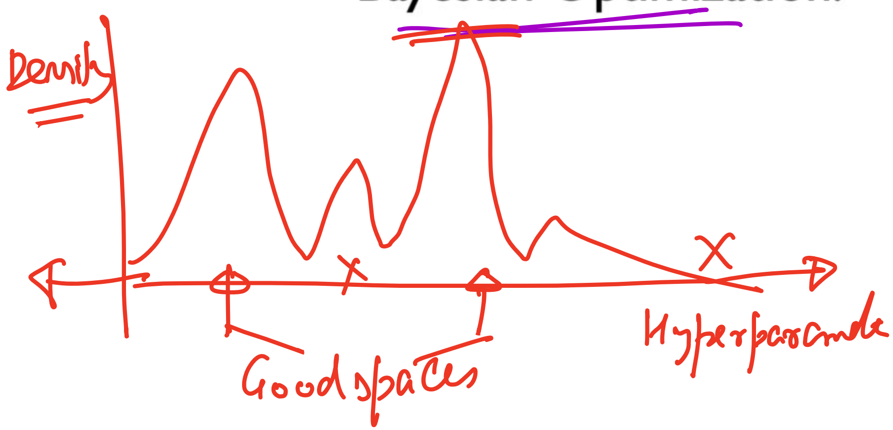
(pick the best on validation data set)

Hyper-parameter tuning methods



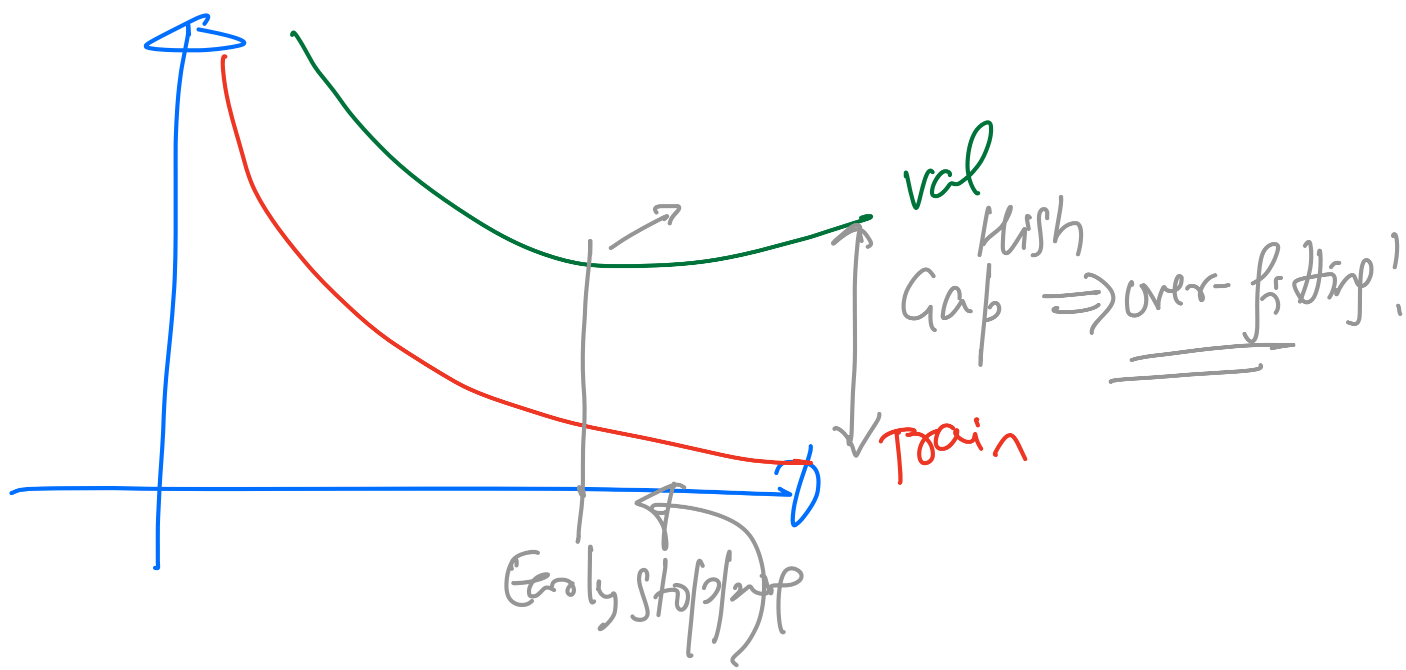
Hyper-parameter tuning methods





How to handle over-fitting in DNNs

1. A DNN model with 100 million parameters and only 100k data points or even a million data points will overfit unless we take care of over-fitting.



Over-fitting in DNNs

How to handle over-fitting in DNNs

1. A DNN model with 100 million parameters and only 100k data points or even a million data points will overfit unless we take care of over-fitting.
2. Weight regularization can help - ℓ_1, ℓ_2

Over-fitting in DNNs

How to handle over-fitting in DNNs

1. A DNN model with 100 million parameters and only 100k data points or even a million data points will overfit unless we take care of over-fitting.
2. Weight regularization can help - ℓ_1, ℓ_2
3. More common over-fitting strategy for DL?

Over-fitting in DNNs

How to handle over-fitting in DNNs

1. A DNN model with 100 million parameters and only 100k data points or even a million data points will overfit unless we take care of over-fitting.
2. Weight regularization can help - ℓ_1, ℓ_2
3. More common over-fitting strategy for DL?
4. Dropouts!

very specific to DL

Over-fitting in DNNs

How to handle over-fitting in DNNs

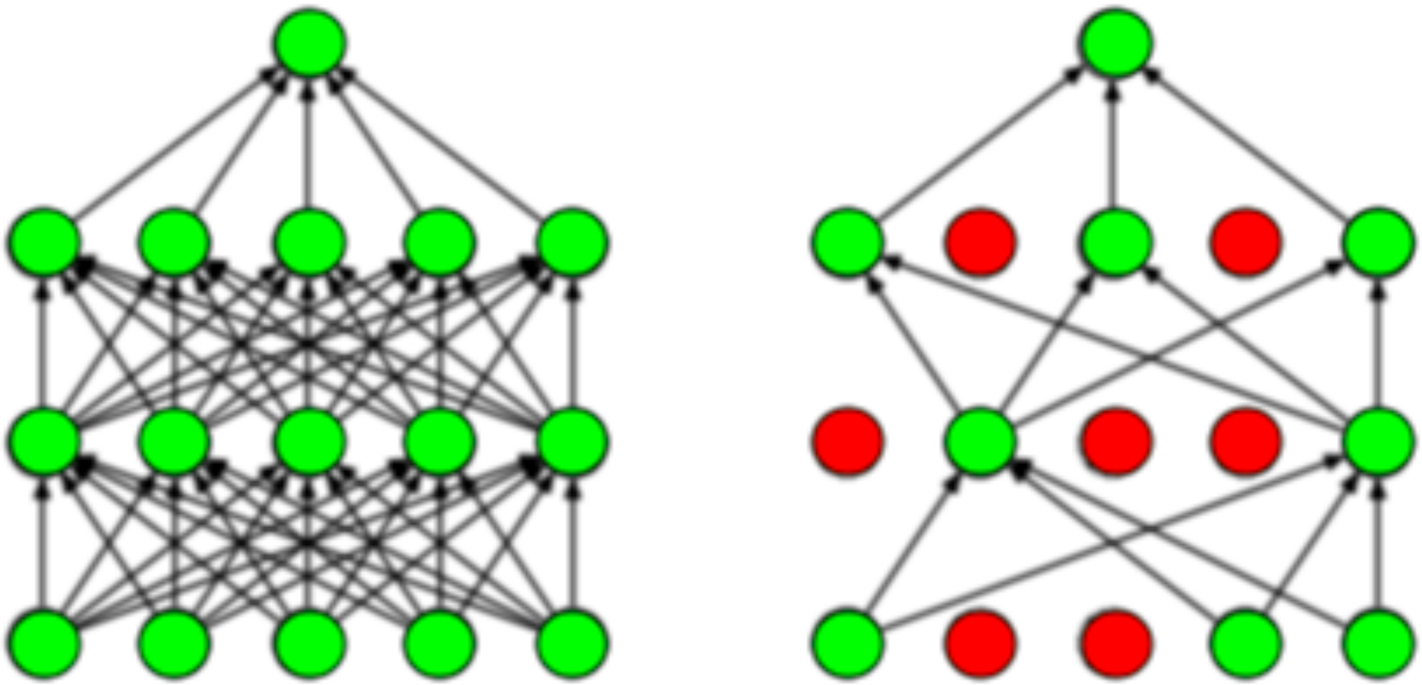
1. A DNN model with 100 million parameters and only 100k data points or even a million data points will overfit unless we take care of over-fitting.
2. Weight regularization can help - ℓ_1, ℓ_2
3. More common over-fitting strategy for DL?
4. Dropouts!
5. Early stopping is also a great strategy! Stop training the DL model when the validation error starts increasing. How's this different from regular validation we were doing earlier??

Over-fitting in DNNs

How to handle over-fitting in DNNs

1. A DNN model with 100 million parameters and only 100k data points or even a million data points will overfit unless we take care of over-fitting.
2. Weight regularization can help - ℓ_1, ℓ_2
3. More common over-fitting strategy for DL?
4. Dropouts!
5. Early stopping is also a great strategy! Stop training the DL model when the validation error starts increasing. How's this different from regular validation we were doing earlier??
6. Book by Yoshua Bengio has tons of details and great reference for Deep Learning! et al

Taking care of Over-fitting: Dropouts



(a) Standard Neural Net

(b) After applying dropout.

Feedforward NN

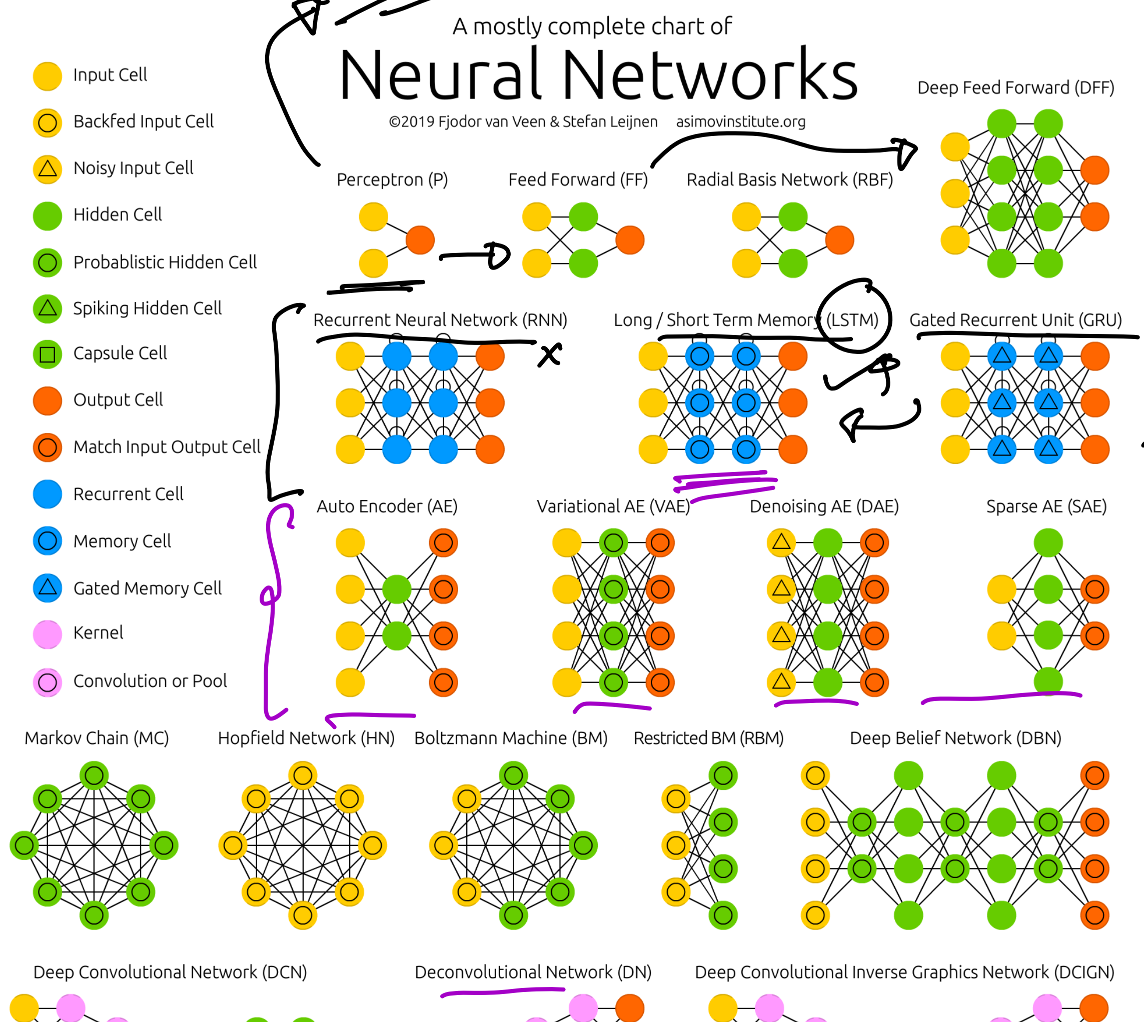
With dropouts
DL can be seen as an ensemble of partially connected NNs

More DL Architectures

Neural Networks Zoo

Zoo Reference

Connected to NLP



Some of the earlier models for sequential NLP tasks

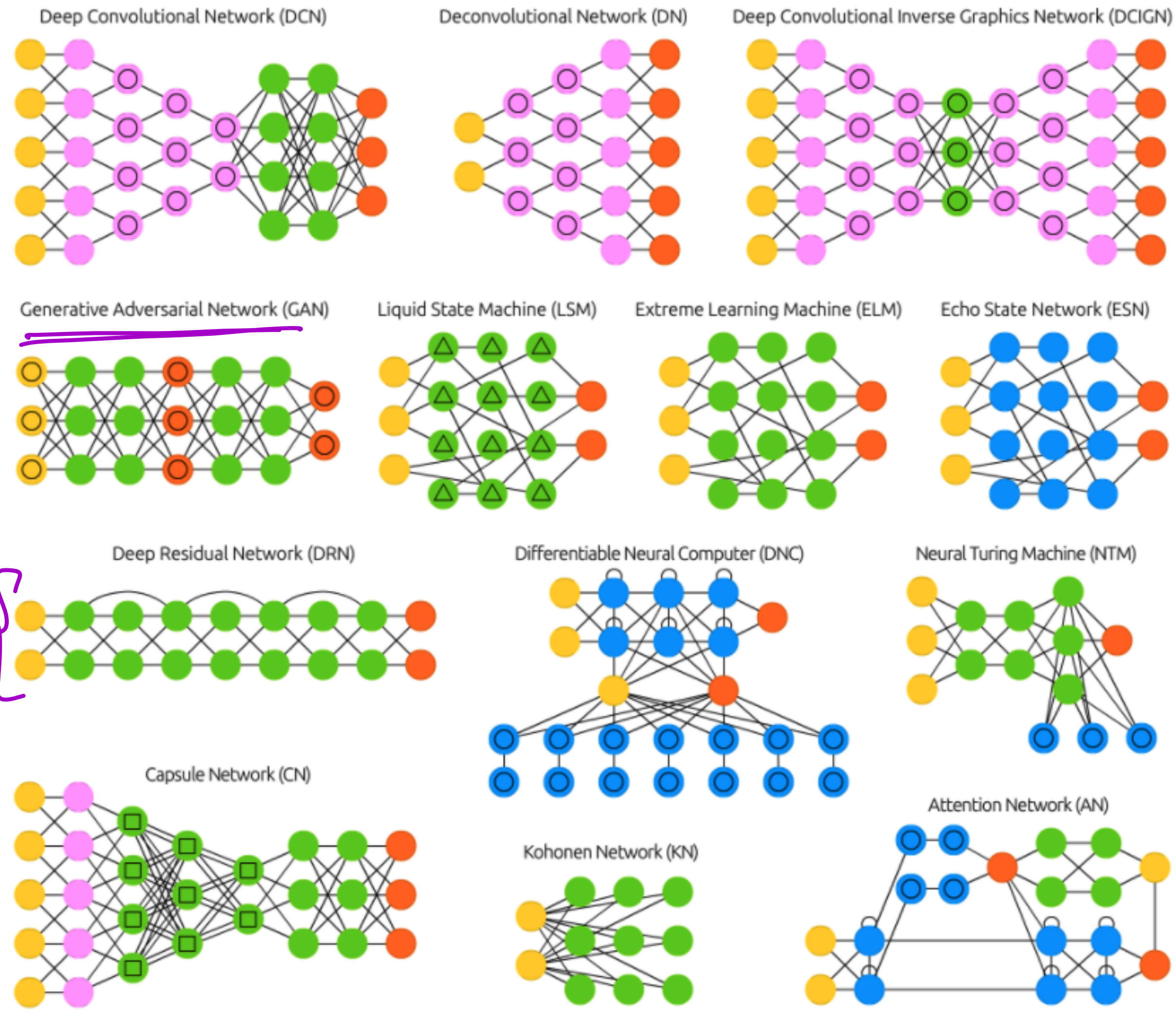
Today

CNN

More DL Architectures

Neural Networks Zoo

↓ CNN (Images)

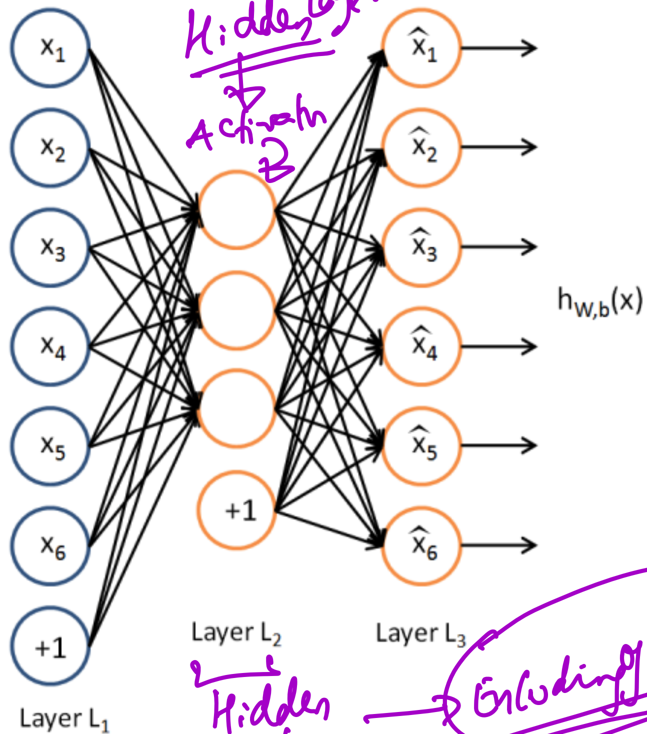


Auto Encoders

Input

Predicted Input

Hidden Layer + Activation



unsupervised / Self-supervised

AE
1. Encoding / Embedding
2. Account for non-Linearity

PCA → Linear Model

Applications
1. Dim Reduction
2. Embeddings
3. Non Linearity
4. De-Noising

Hidden Layer → Encoding the Input

Predict the Input

Simple AE :- Input Layer + Hidden Layer + output Layer

ICE #2

PCA vs Auto Encoder

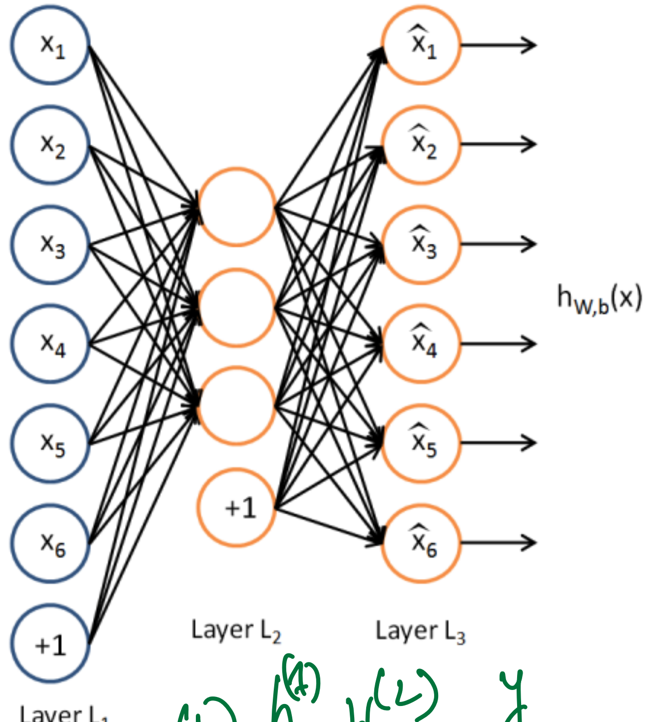
Which of the following statements are true ?

1. Both PCA and Auto Encoders serve the purpose of dimensionality reduction
2. They are both linear models but one uses a neural nets architecture and the other is based on projections
3. PCA is robust to outliers while Auto Encoders are not
4. Auto Encoders are as better than Glove Embeddings to find low-dim embeddings for words

PCA vs Auto-Encoders

vanilla computation

Baseline/starter (not customized to your problem)



$\hat{x} \approx x$

Linear

$x \quad W^{(1)} \quad h^{(1)} \quad W^{(2)} \quad y$

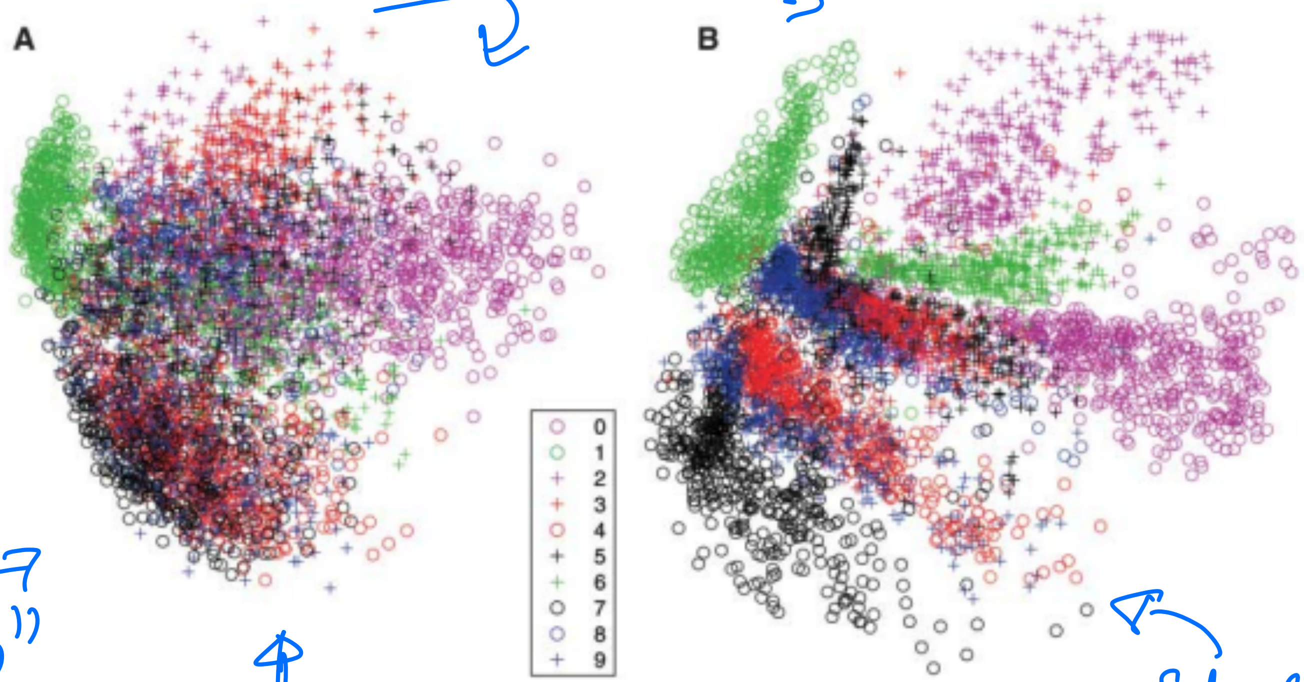
Non-linear activation

$$(y =)\ \hat{x} = g_2\left(W^{(2)} \left(g_1\left(W^{(1)} x\right) + b_1 \right)\right) \, !$$

Reading Reference for AE Dimensionality Reduction

N=500 per Class

Fig. 3. (**A**) The two-dimensional codes for 500 digits of each class produced by taking the first two principal components of all 60,000 training images. (**B**) The two-dimensional codes found by a 784-1000-500-250-2 autoencoder. For an alternative visualization, see (*8*).



"Mixed up"

PCA

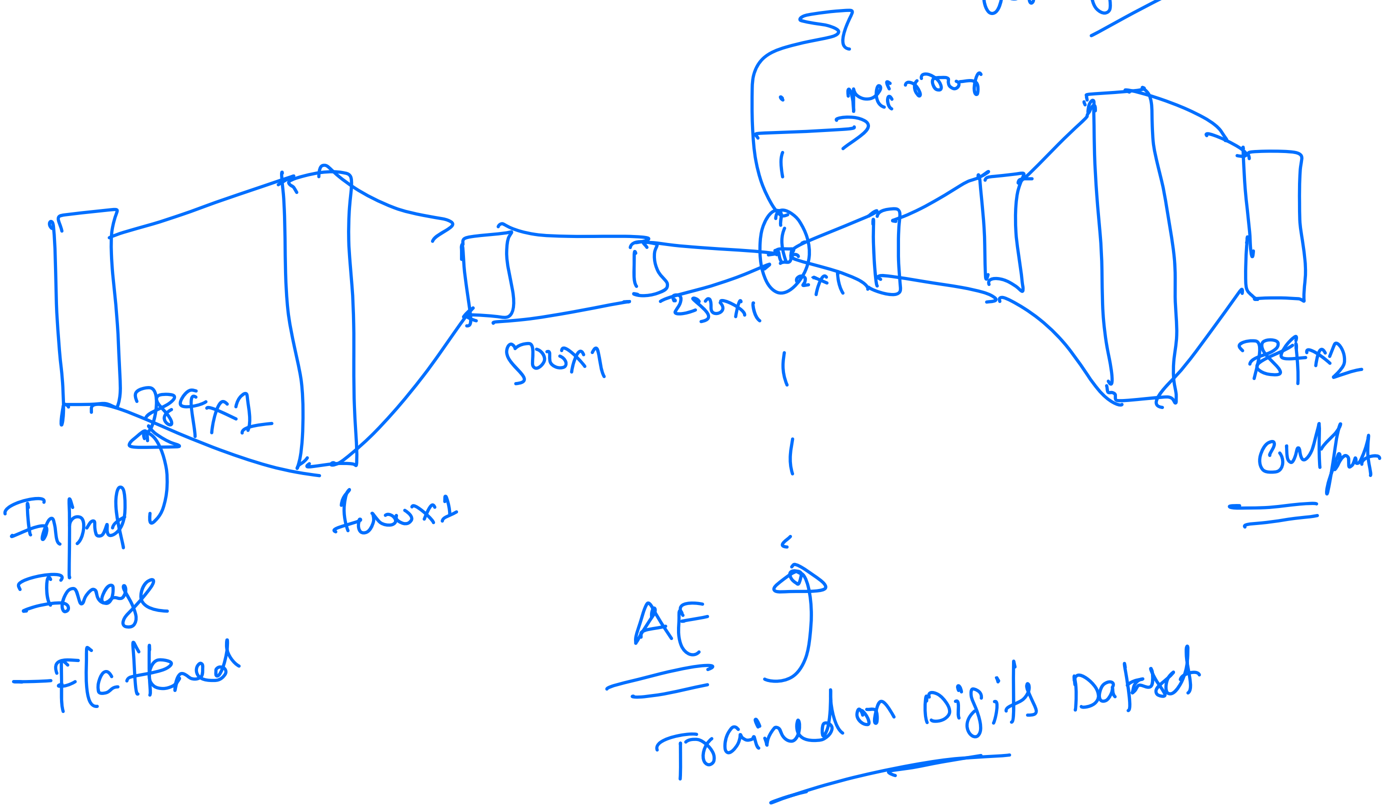
Digits Dataset

AE

Separation of classes

784 - 1000 - 500 - 250 - 2

"2 dim. code" used for visualization

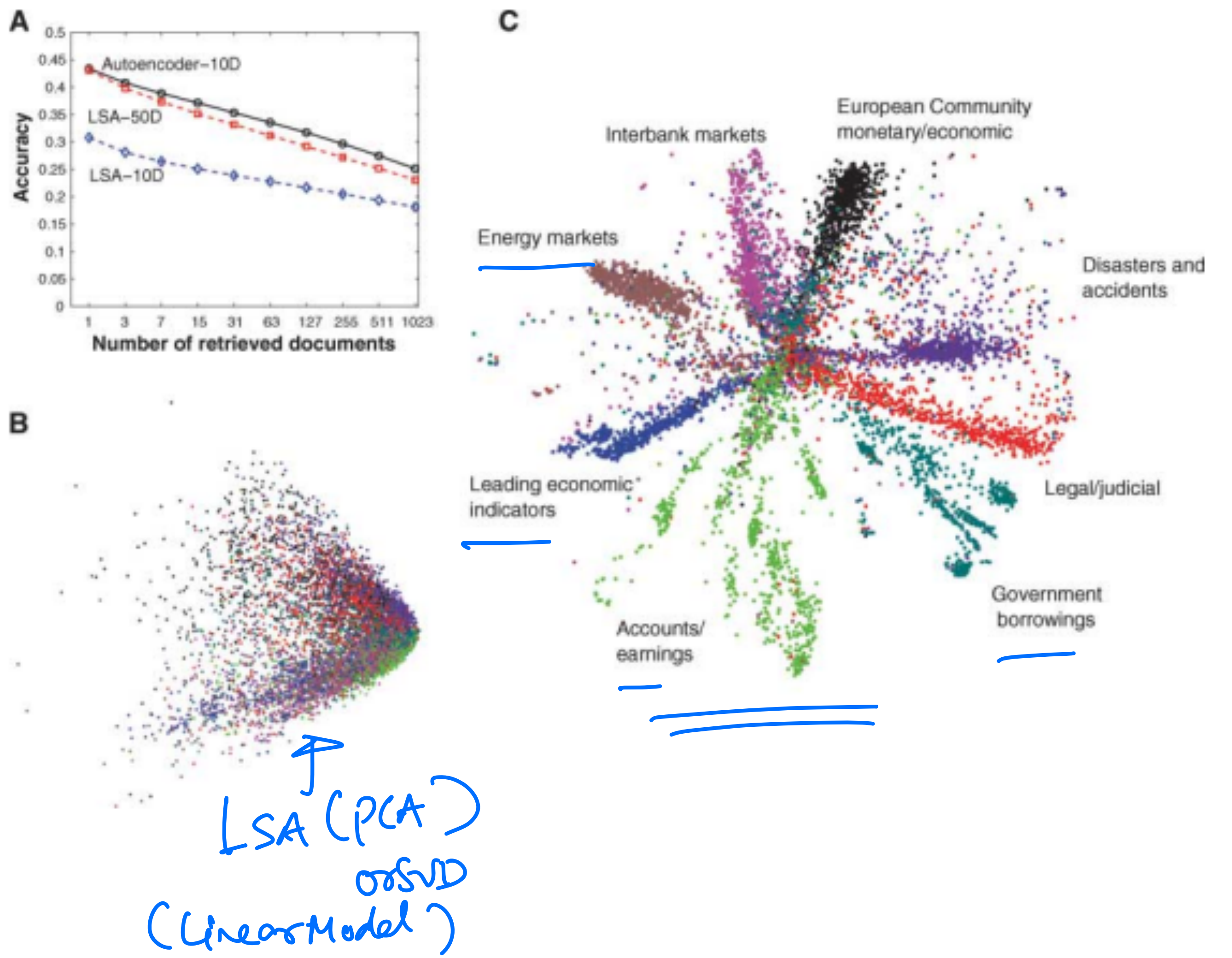


AutoEncoders and Dimensionality Reduction

Reading Reference for AE Dimensionality Reduction

Fig. 4. **(A)** The fraction of retrieved documents in the same class as the query when a query document from the test set is used to retrieve other test set documents, averaged over all 402,207 possible queries. **(B)** The codes produced by two-dimensional LSA. **(C)** The codes produced by a 2000-500-250-125-2 autoencoder.

Code



AutoEncders Summary

1. Auto-Encoders are a method for dimensionality reduction and can do better than PCA for visualization

AutoEncders Summary

1. Auto-Encoders are a method for dimensionality reduction and can do better than PCA for visualization
2. Use Neural Networks architecture and hence can encode non-linearity in the embeddings

AutoEncders Summary

1. Auto-Encoders are a method for dimensionality reduction and can do better than PCA for visualization
2. Use Neural Networks architecture and hence can encode non-linearity in the embeddings
3. AEs can learn non-linear embeddings for data in a self-supervised manner!

AutoEncders Summary

1. Auto-Encoders are a method for dimensionality reduction and can do better than PCA for visualization
2. Use Neural Networks architecture and hence can encode non-linearity in the embeddings
3. AEs can learn non-linear embeddings for data in a self-supervised manner!
4. Can be a starting point to extract concise feature embeddings for a supervised learning model
5. Anything else? → De-Noising

AutoEncders Summary

1. Auto-Encoders are a method for dimensionality reduction and can do better than PCA for visualization
2. Use Neural Networks architecture and hence can encode non-linearity in the embeddings
3. AEs can learn non-linear embeddings for data in a self-supervised manner!
4. Can be a starting point to extract concise feature embeddings for a supervised learning model
5. Anything else?
6. Auto Encoders can learn convolutional layers instead of dense layers - Better for images! More flexibility!!

Removing obstacles in images

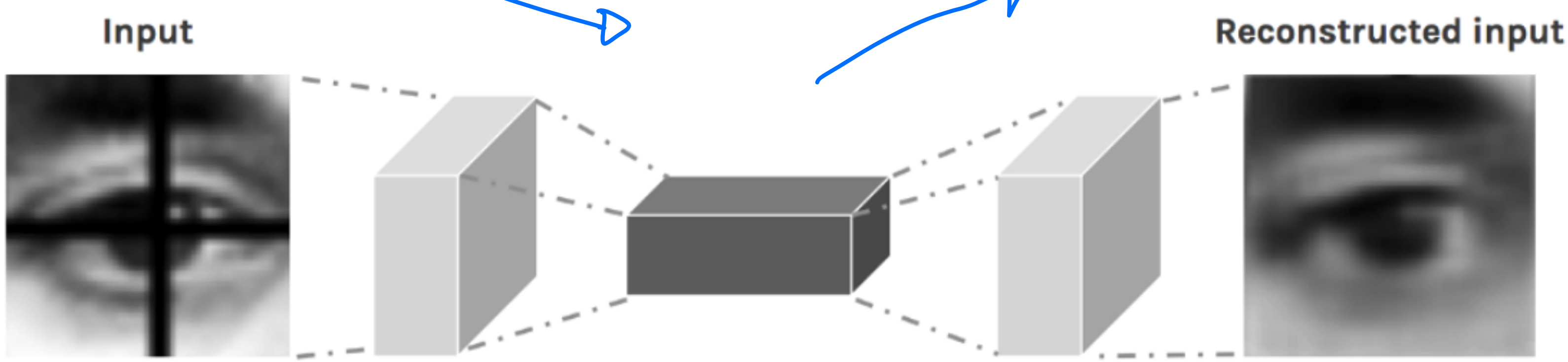


Figure 12: Reconstructed image from missing image [14]

Noisy Input

Input

Removing obstacles in images

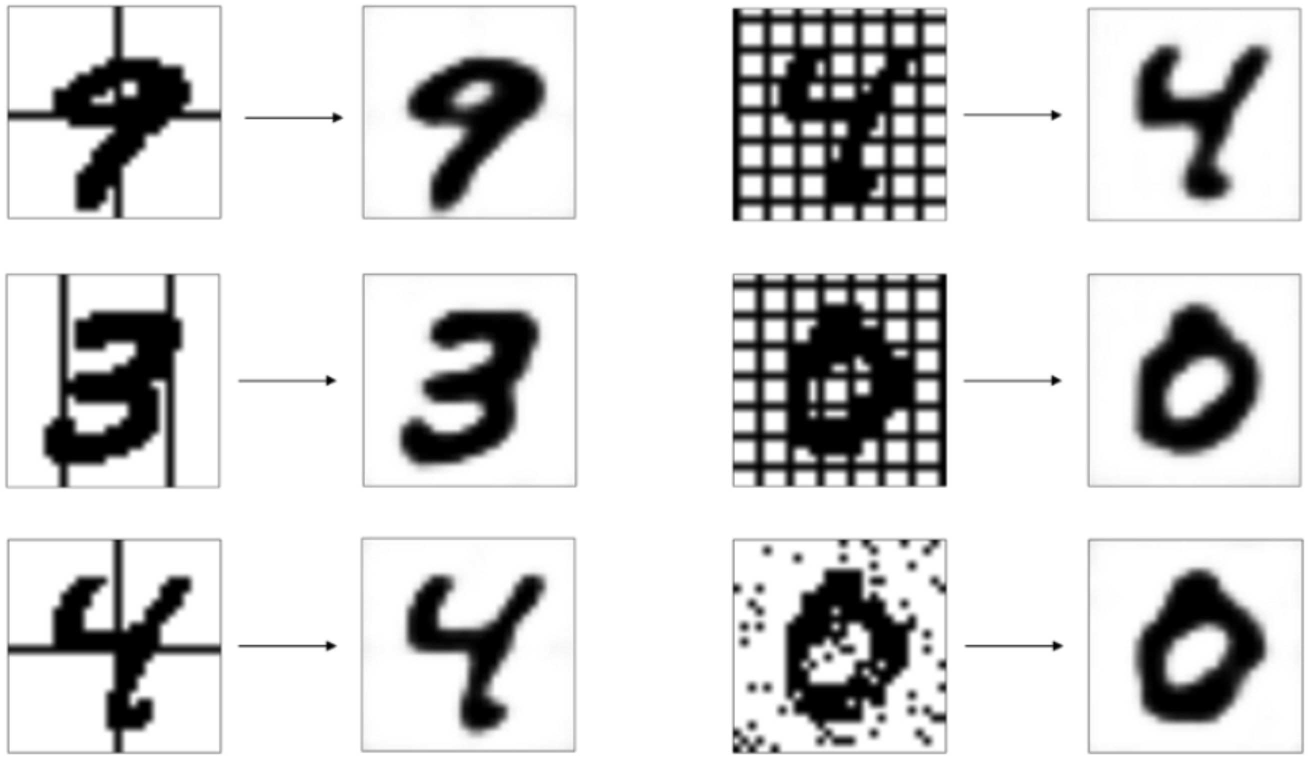
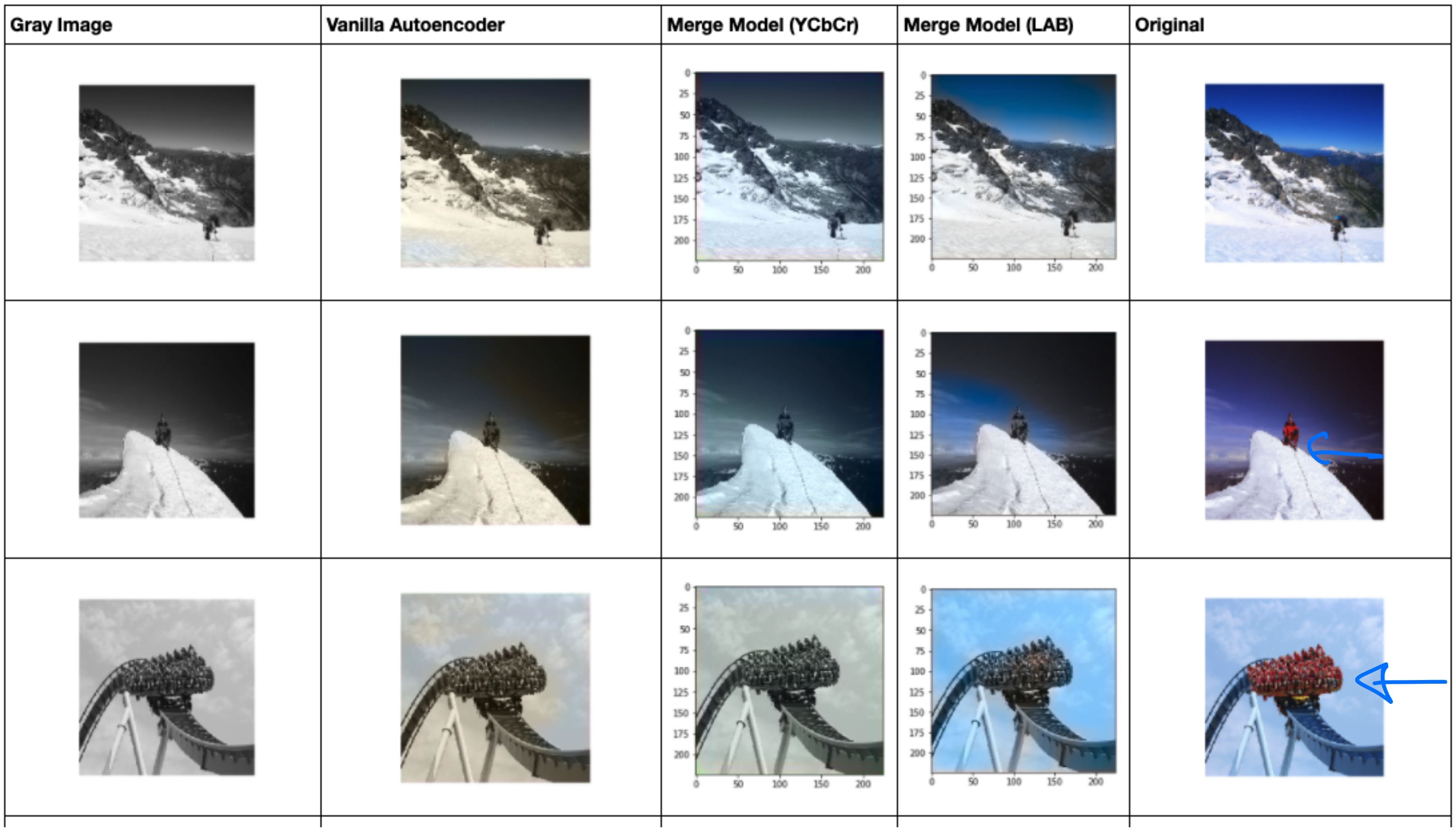


Figure 13: Source [15]

Specific ("sparse noise")

Random noise

Coloring Images



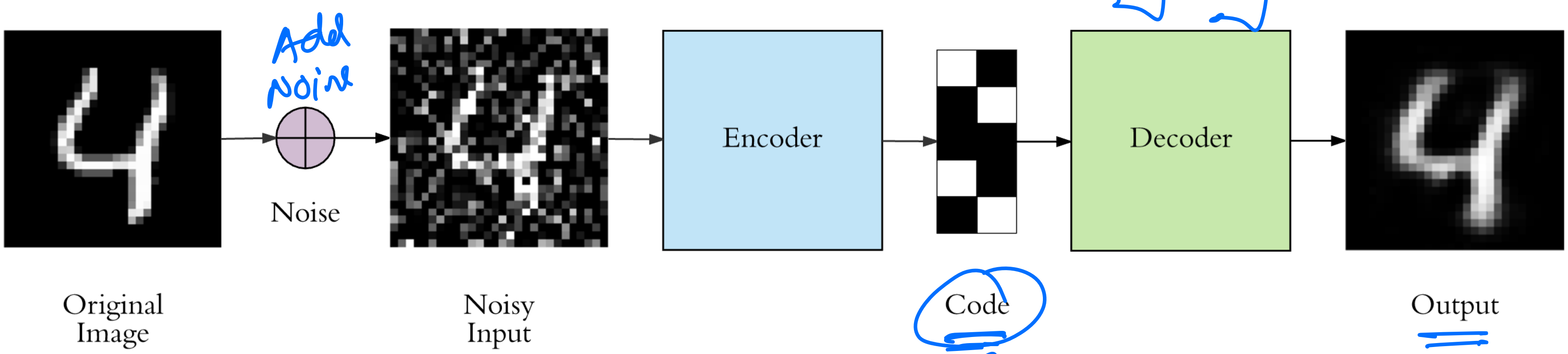
Input:- B&W Image

output:- Color version of Image

works for certain settings:-
e.g. Nature Images with in it

De-noising Auto Encoders

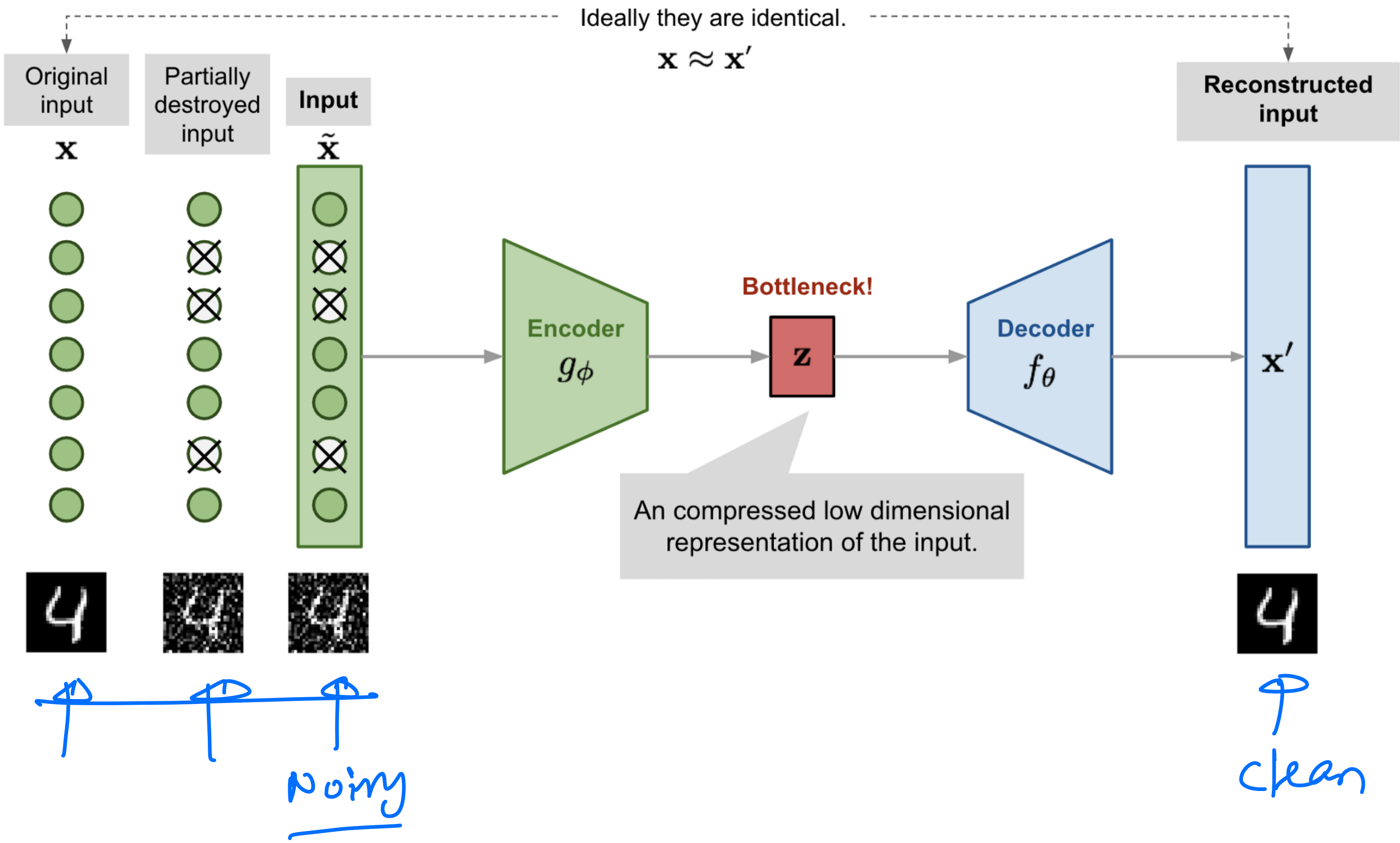
Noisy Image



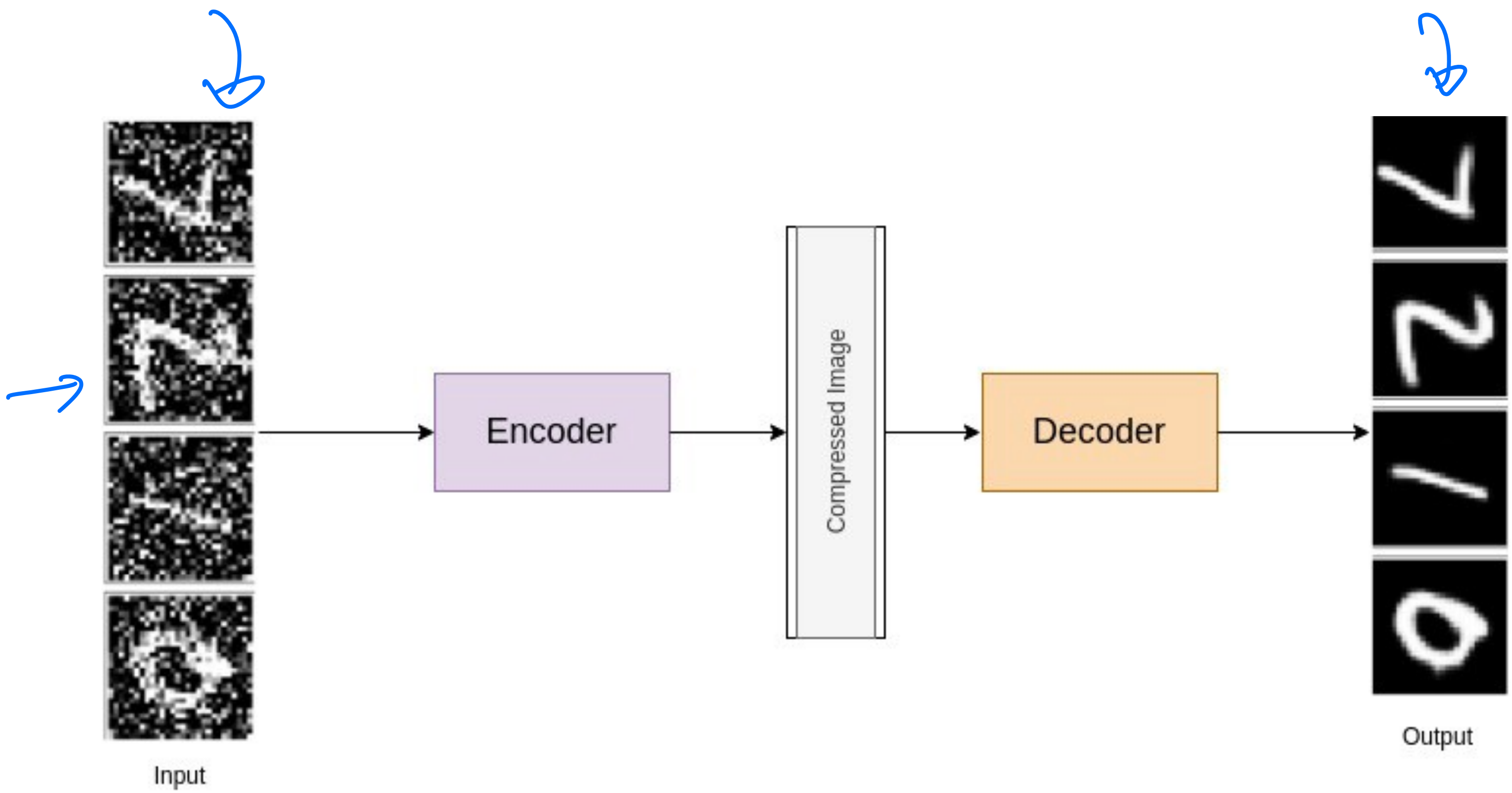
Embeddings"

"Noise-Aware

De-noising Auto Encoders



De-noising Auto Encoders



De-noising Auto Encoders

Details

- Just like an Auto Encoder

De-noising Auto Encoders

Details

- Just like an Auto Encoder
- Difference: Noise is injected in the inputs on purpose but output is a clean data point.

De-noising Auto Encoders

Details

- Just like an Auto Encoder
- Difference: Noise is injected in the inputs on purpose but output is a clean data point.
- This forces the Auto Encoder to “de-noise” data, esp. useful for images!

De-noising Auto Encoders

Details

- Just like an Auto Encoder
- Difference: Noise is injected in the inputs on purpose but output is a clean data point.
- This forces the Auto Encoder to "de-noise" data, esp. useful for images!
- Esp. useful for a category of objects or images (e.g. digit recognition or face recognition, etc)

De-noising Auto Encoders

Details

- Just like an Auto Encoder
- Difference: Noise is injected in the inputs on purpose but output is a clean data point.
- This forces the Auto Encoder to "de-noise" data, esp. useful for images!
- Esp. useful for a category of objects or images (e.g. digit recognition or face recognition, etc)
- De-noising AEs can be used to learn **noise-aware embeddings** - Helps with improving robustness of downstream models

ICE #3

Unsupervised Learning

Which of these is NOT an example of unsupervised learning?

1. Perceptron
2. Auto Encoder
3. De-noising Auto Encoder
4. K-means++
5. None of the above
6. All of the above

AutoEncoder Tensorflow Tutorial

AutoEncoder TensorFlow Tutorial

Breakouts Time 1

5 mins

Discuss in your groups what are some real-world applications of any or many of the Auto Encoder Architectures we discussed so far you can think of in your area of work or in a standard context e.g. images.

Sequence structure in NLP

Example

I love this car! Positive Sentiment

Sequence structure in NLP

Example

I love this car! Positive Sentiment

Example

I am not sure I love this car! Negative Sentiment

Sequence structure in NLP

Example

I love this car! Positive Sentiment

Example

I am not sure I love this car! Negative Sentiment

Example

I don't think its a bad car at all! $\rightarrow$ Positive Sentiment

Sequence structure in NLP

Example

I love this car! Positive Sentiment

Example

I am not sure I love this car! Negative Sentiment

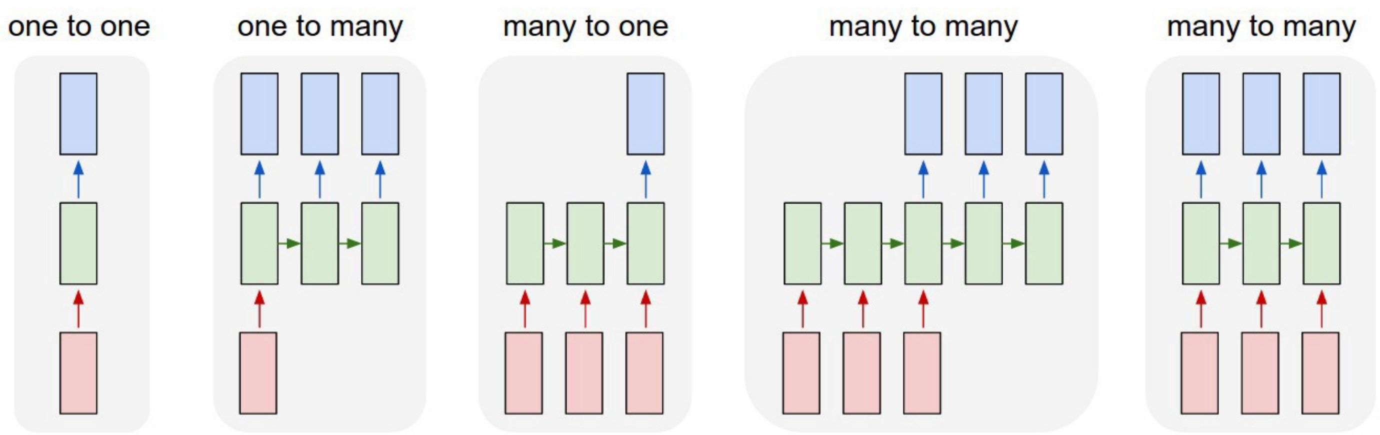
Example

I don't think its a bad car at all! $\rightarrow$ Positive Sentiment

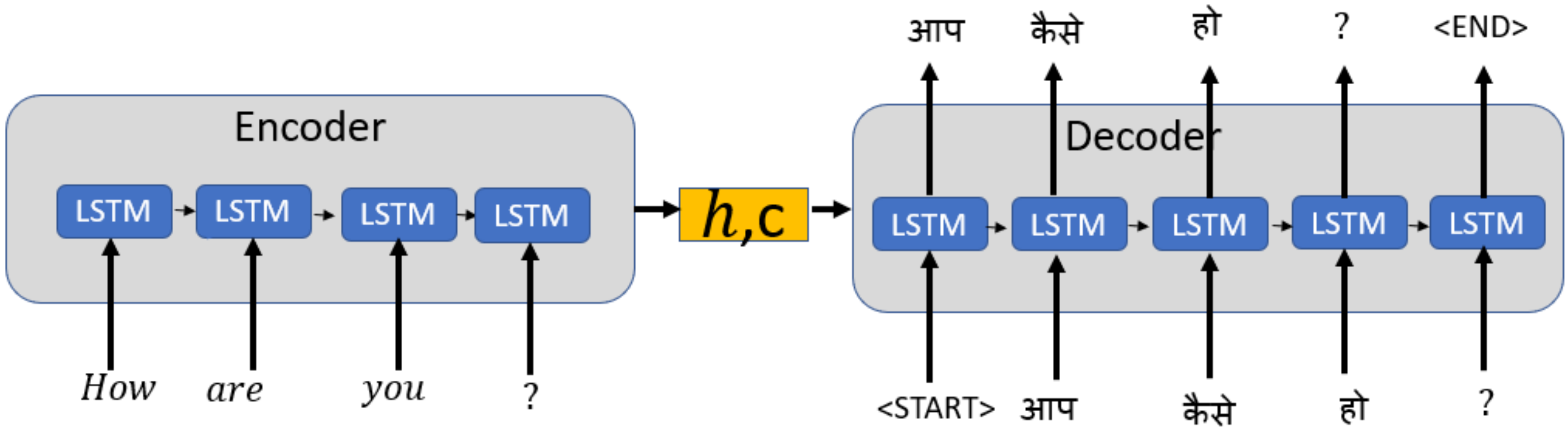
Example

Have to carry the **context(state)** from some-time back to fully understand what's happening!

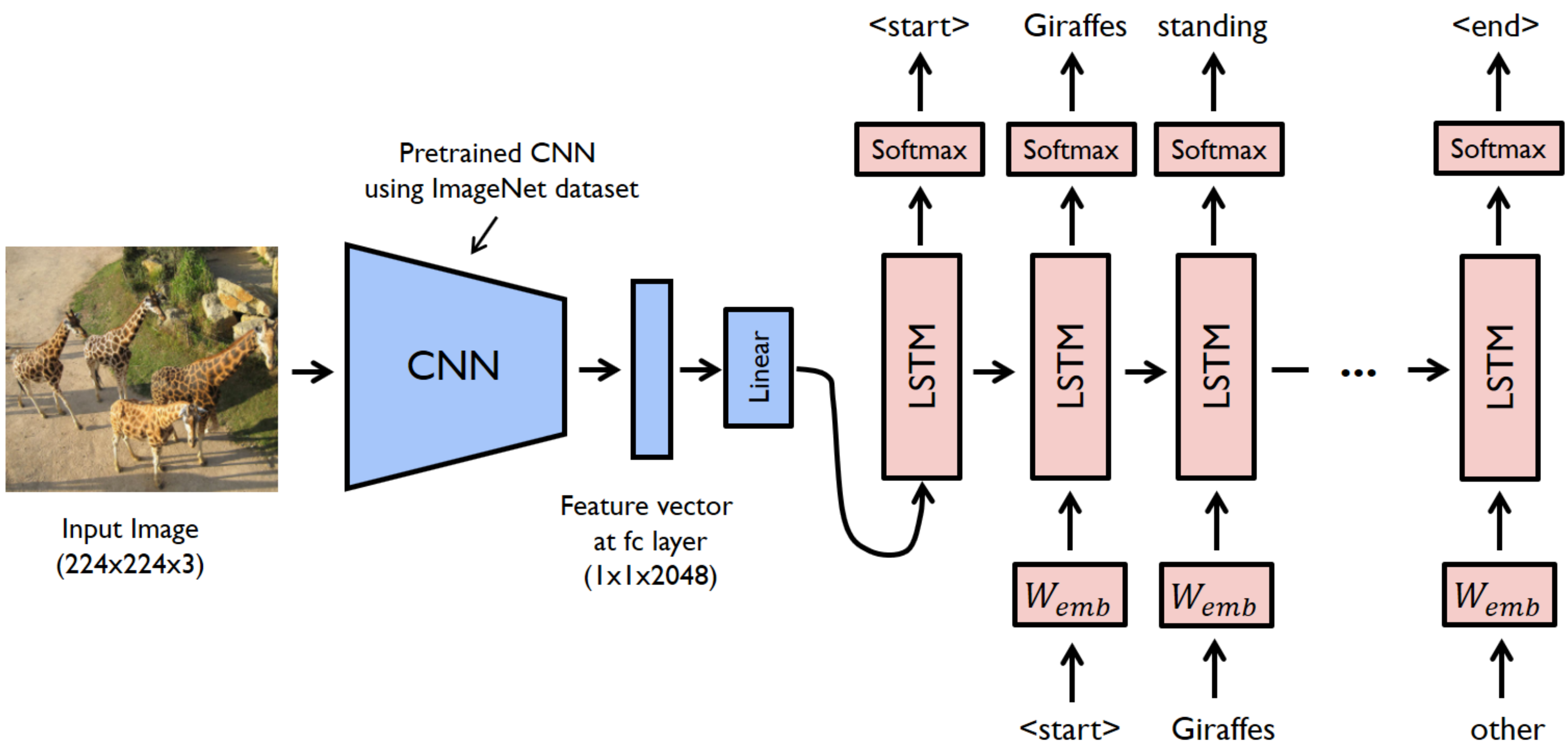
Sequence to Sequence Model (LSTM) Applications



Sequence to Sequence Model (LSTM) Applications



Sequence to Sequence Model (LSTM) Applications



Breakouts Time #2

Auto-complete — 5 mins

Let's say you are tasked with building an in-email auto-completion application, which can help complete partial sentences into full sentences through suggestions (auto-complete). How would you use what we have learned so far to model this? What architecture would you use? What would be your data? And what are some pitfalls or painpoints your model should address?

Applications in Natural Language Processing (NLP)

Applications

1. Topic Modeling

Applications in Natural Language Processing (NLP)

Applications

1. Topic Modeling
2. Machine Translation/Language Translation

Applications in Natural Language Processing (NLP)

Applications

1. Topic Modeling
2. Machine Translation/Language Translation
3. Sentiment Analysis

Applications in Natural Language Processing (NLP)

Applications

1. Topic Modeling
2. Machine Translation/Language Translation
3. Sentiment Analysis
4. Chat bots

Applications in Natural Language Processing (NLP)

Applications

1. Topic Modeling
2. Machine Translation/Language Translation
3. Sentiment Analysis
4. Chat bots
5. Document Summarization

Applications in Natural Language Processing (NLP)

Applications

1. Topic Modeling
2. Machine Translation/Language Translation
3. Sentiment Analysis
4. Chat bots
5. Document Summarization
6. Many more!

Extra Slides

Topic Modeling

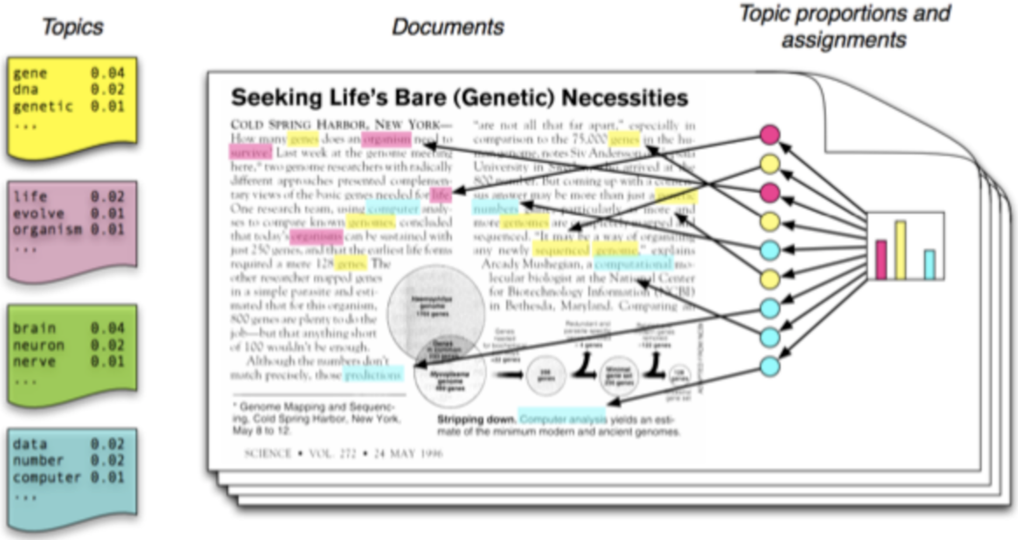
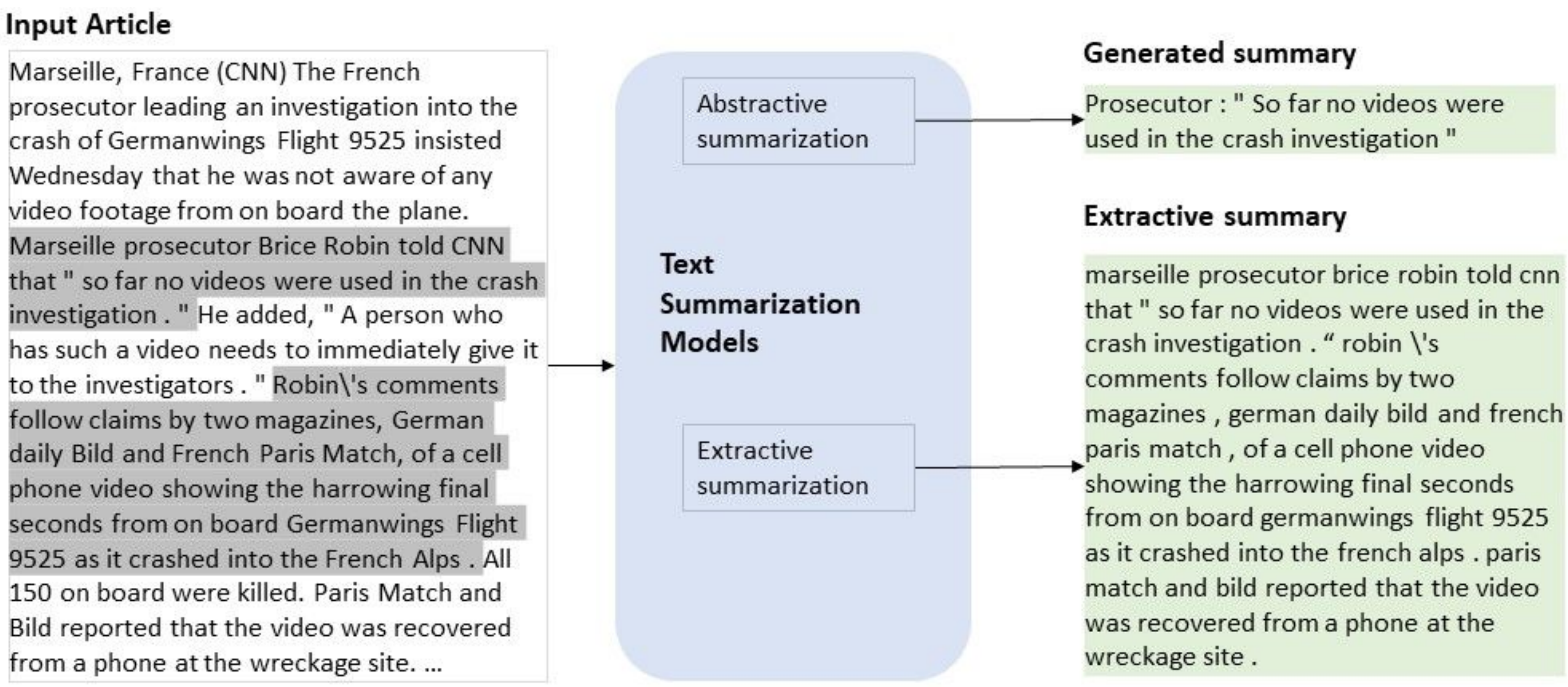


Figure source: Blei, D. M. (2012). Probabilistic topic models. *Communications of the ACM*, *55*(4), 77-84.

Document Summarization



Document Summarization — Extractive

Evaluation Metrics

1. ROUGE score: Recall-Oriented Understudy for Gisting Evaluation
2. ROUGE-N: N-gram overlap between two summaries

ICE #4

ROUGE-1

Consider the truth summary and an automated summary of an article from International Geographic! Find the ROUGE-N score based on finding the proportion of N-grams in the truth summary that are also in the automated summary for $N = 1$.

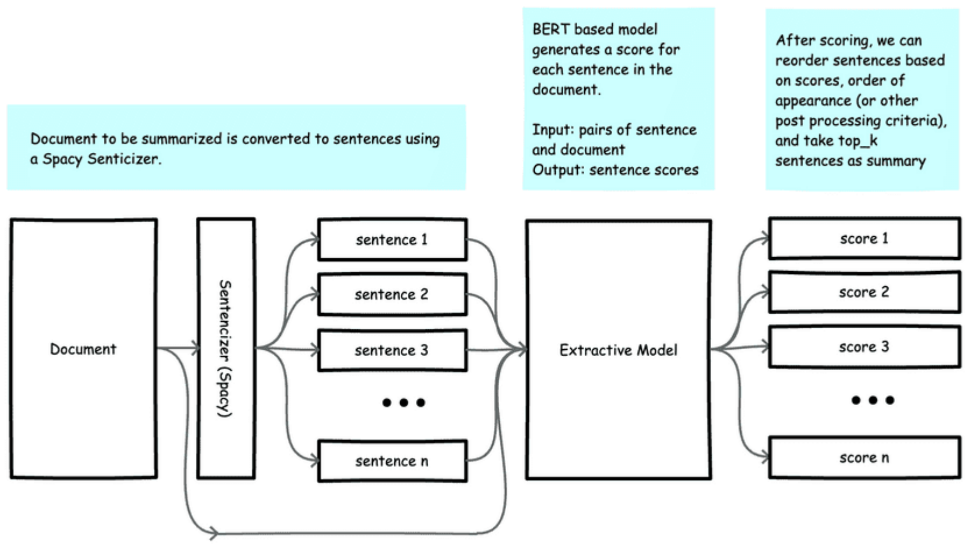
Truth Summary: A symbiotic relationship exists between these two species. The cows feed on wild grass and the egrets feed on the tics found on the surface of the cows.

Automated Summary: These two species have a symbiotic relationship.

ROUGE-1 $=$

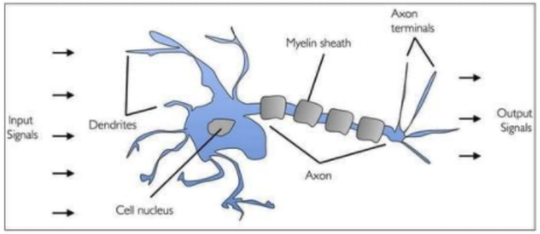
a) 0.33 b) 0.4 c) 0.2 d) 0.25

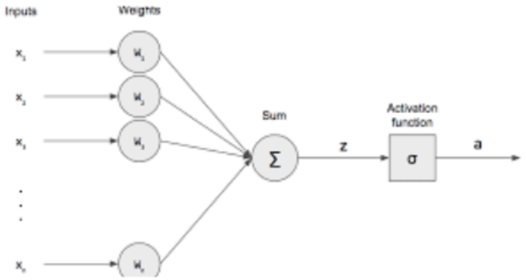
Document Summarization



Evolution of DNN architectures for NLP!

Perceptron

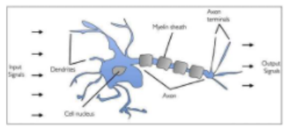


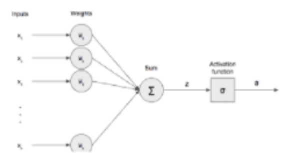


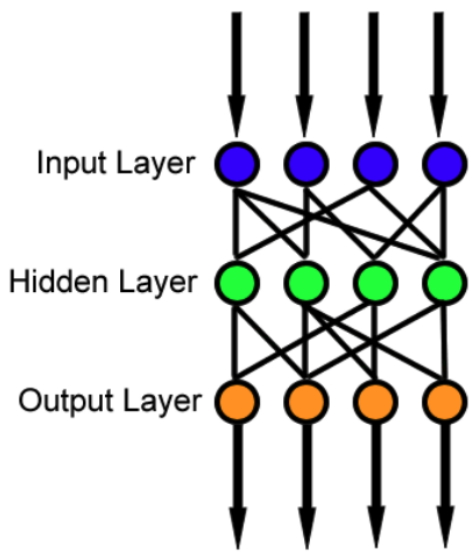
Evolution of DNN architectures for NLP!

Perceptron

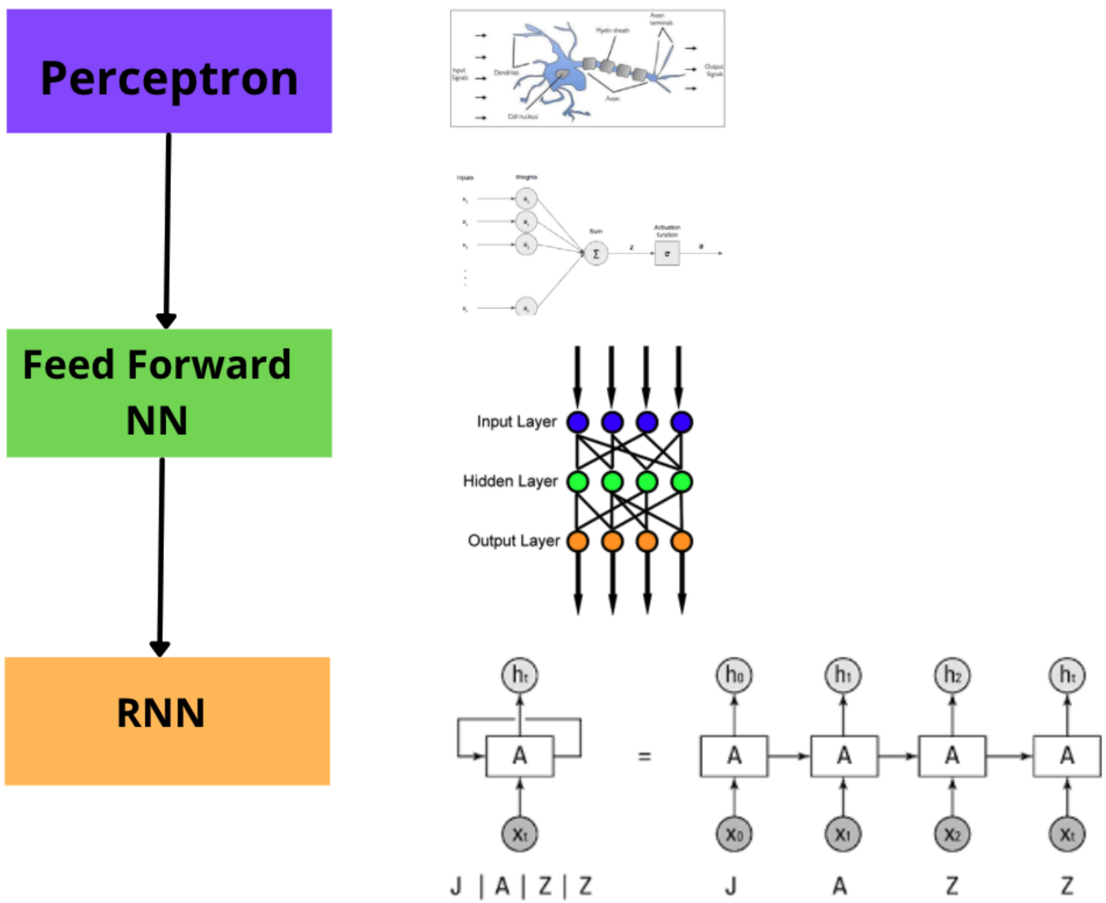
Feed Forward NN



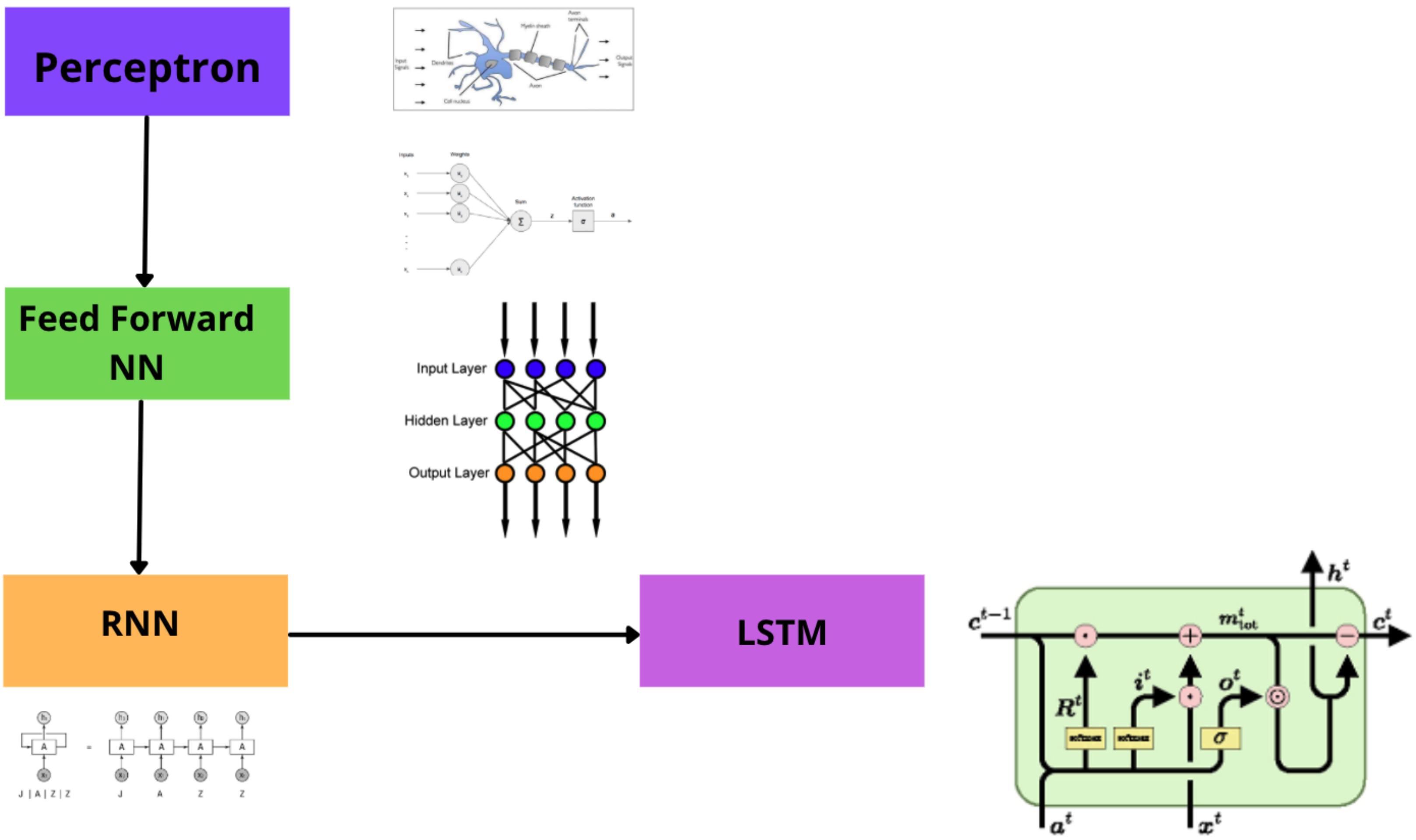




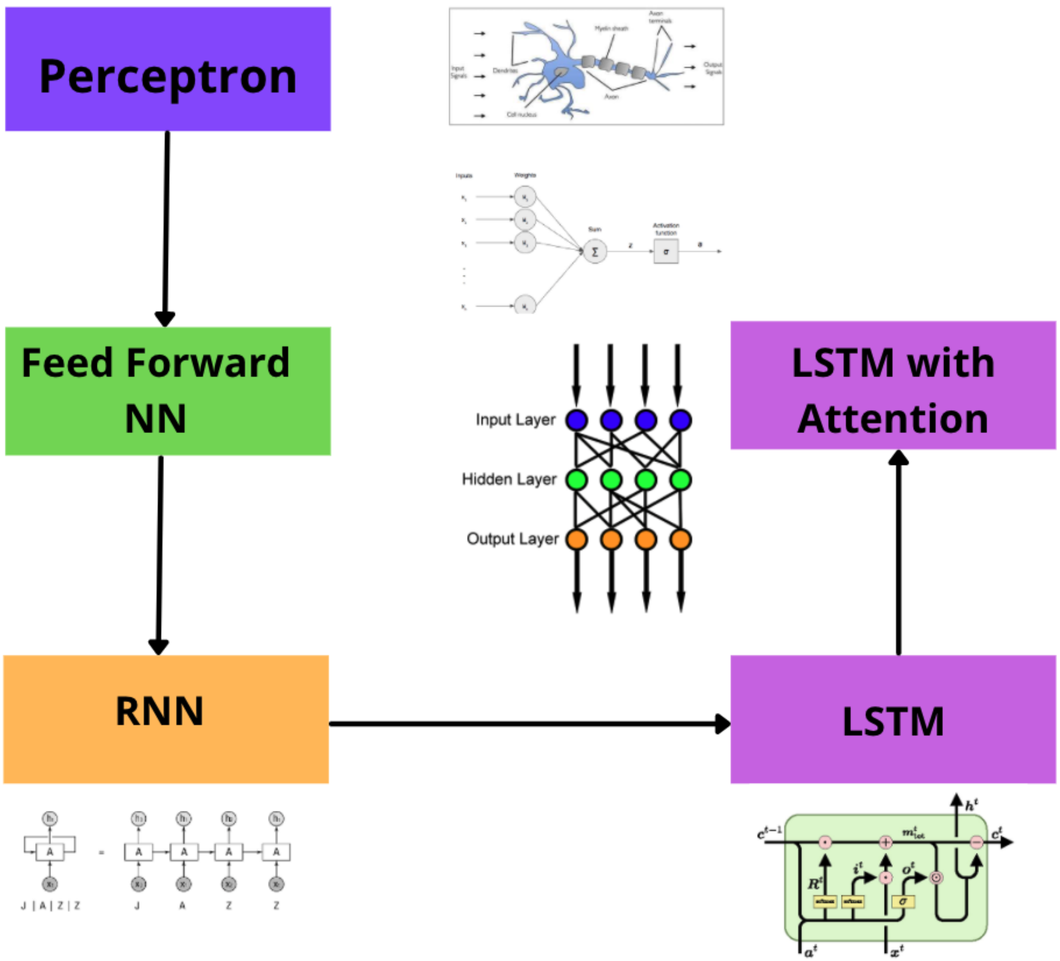
Evolution of DNN architectures for NLP!



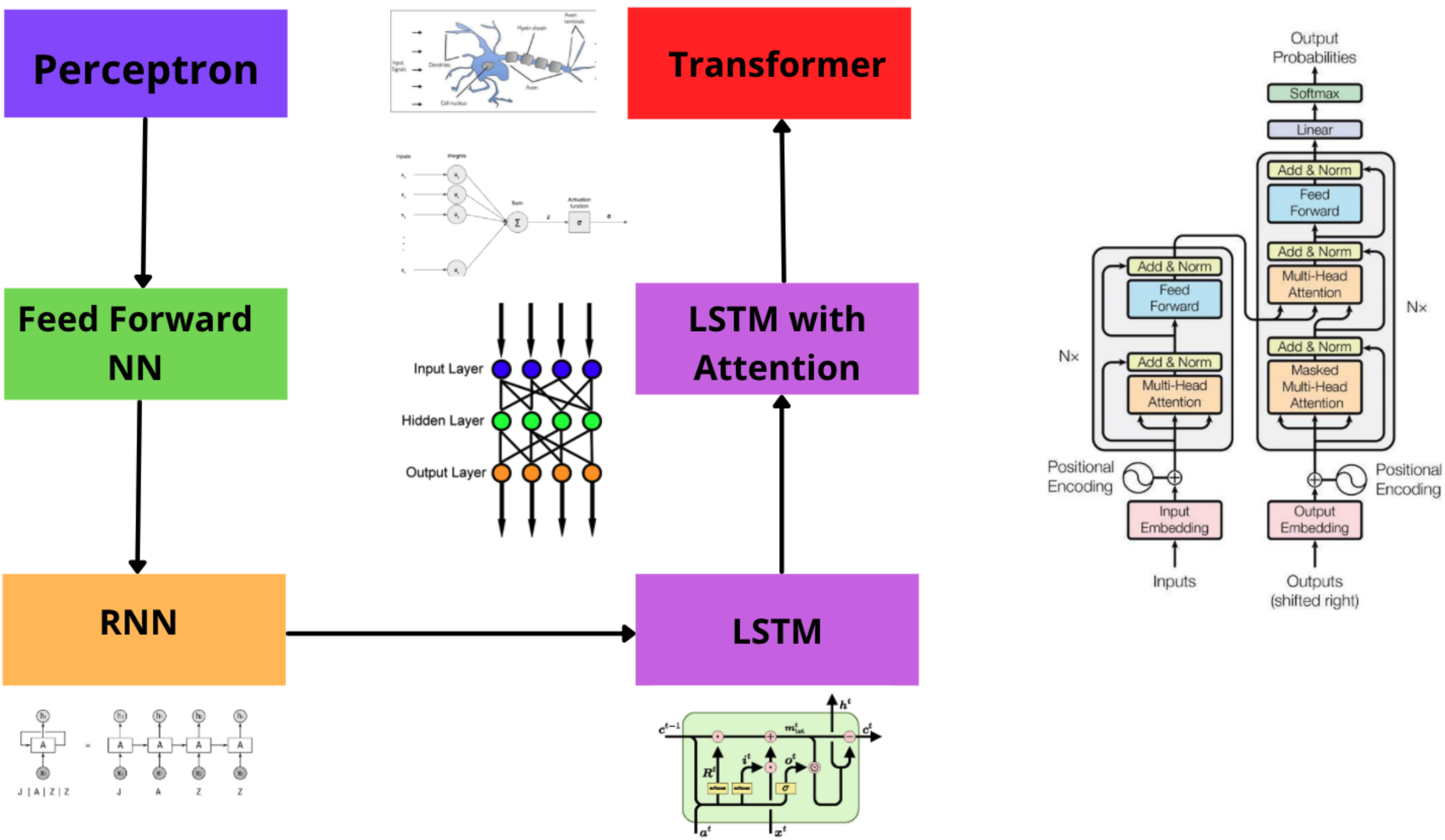
Evolution of DNN architectures for NLP!



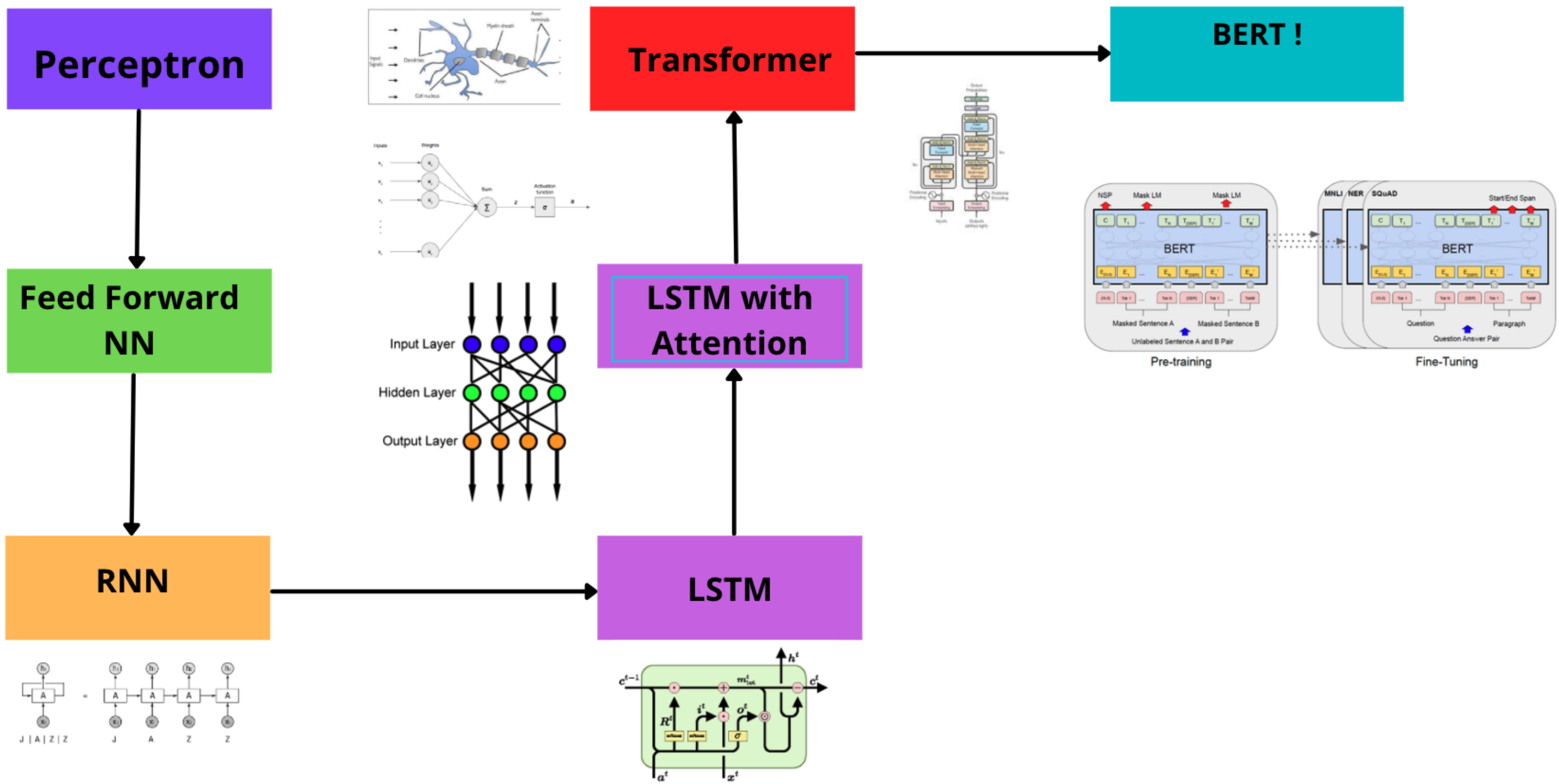
Evolution of DNN architectures for NLP!



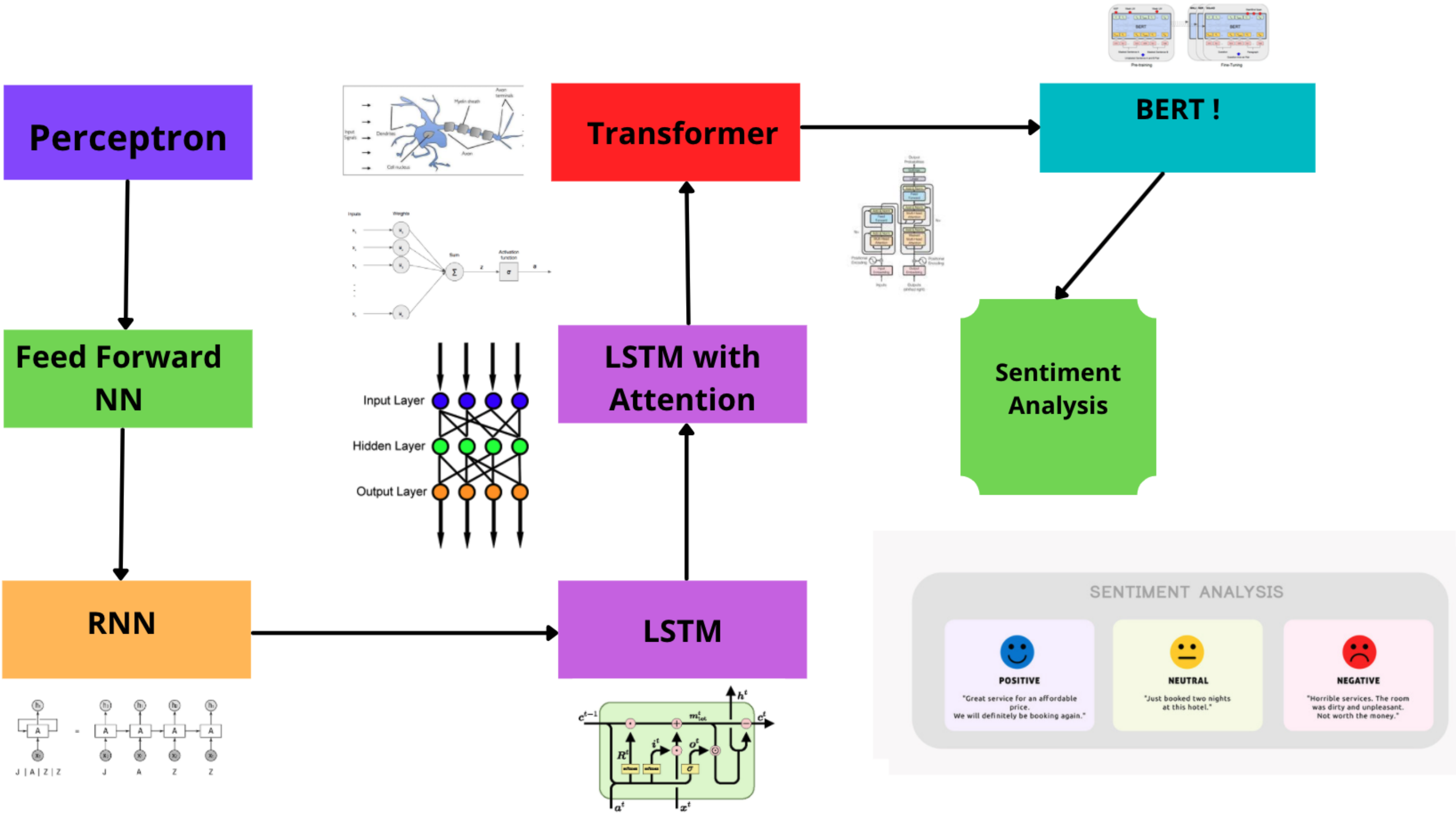
Evolution of DNN architectures for NLP!



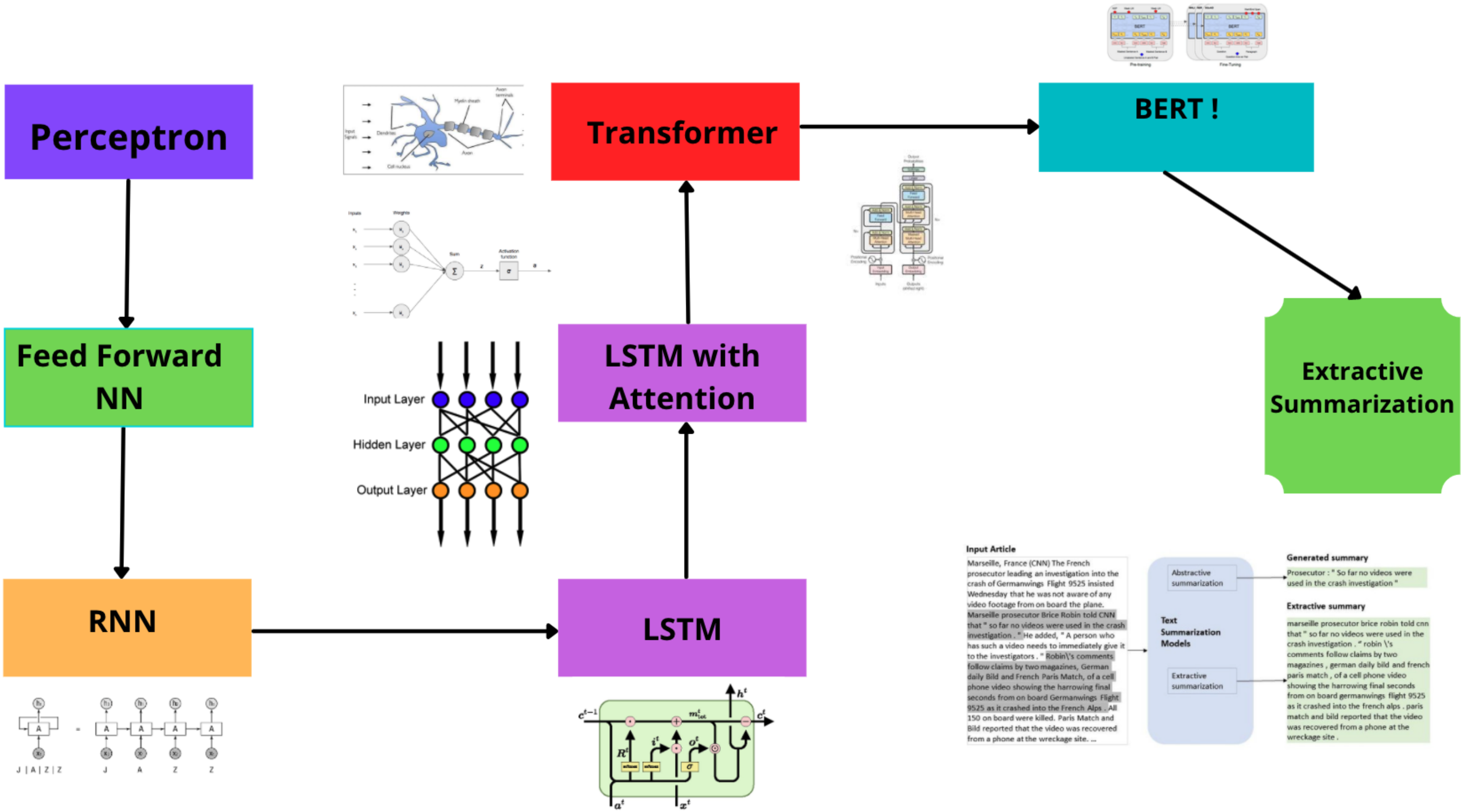
Evolution of DNN architectures for NLP!



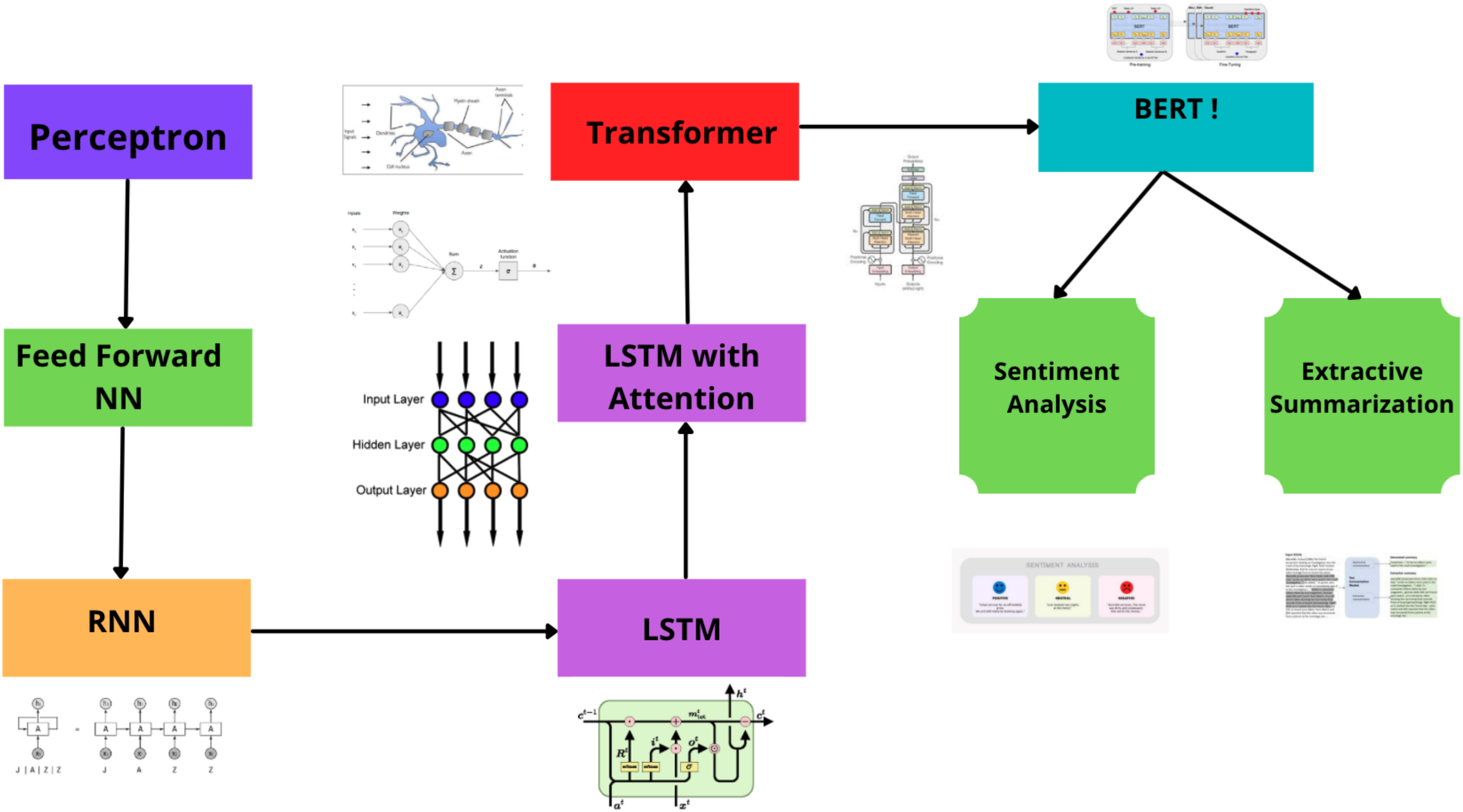
Evolution of DNN architectures for NLP!



Evolution of DNN architectures for NLP!



Evolution of DNN architectures for NLP!



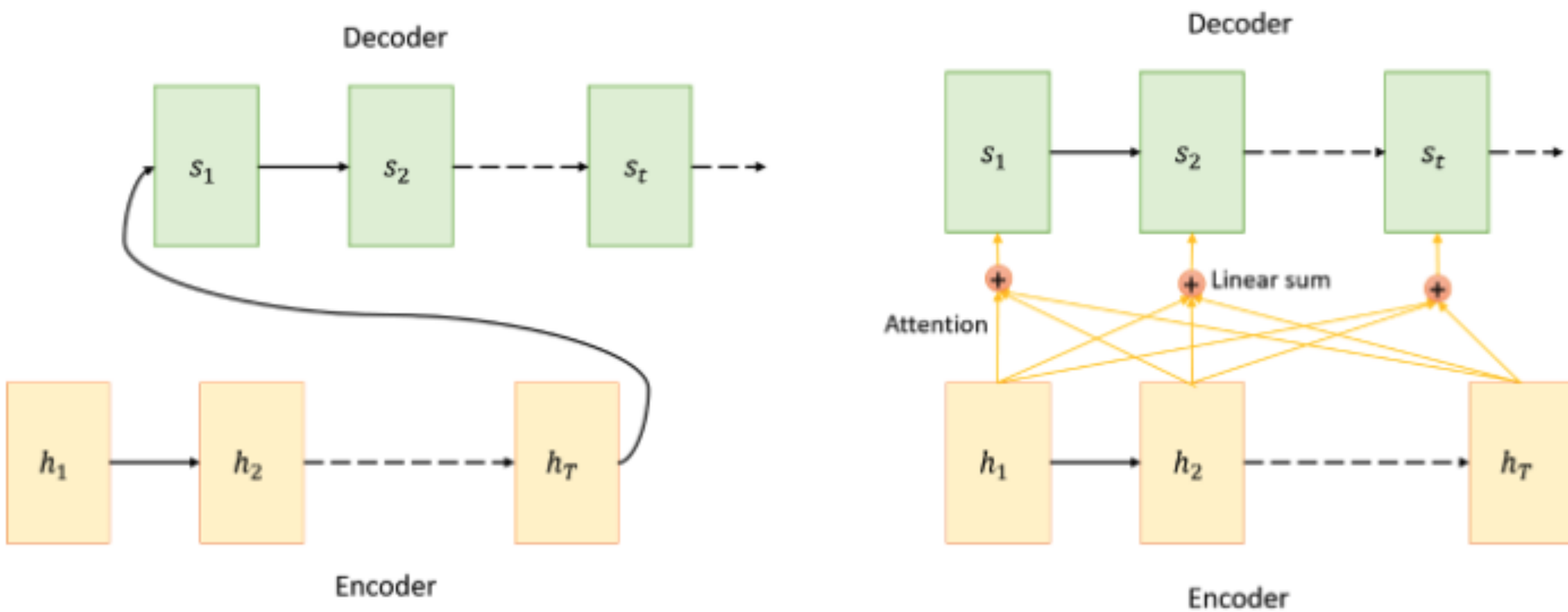
ICE #5

RNN vs LSTM

Which of the following statements are NOT true?

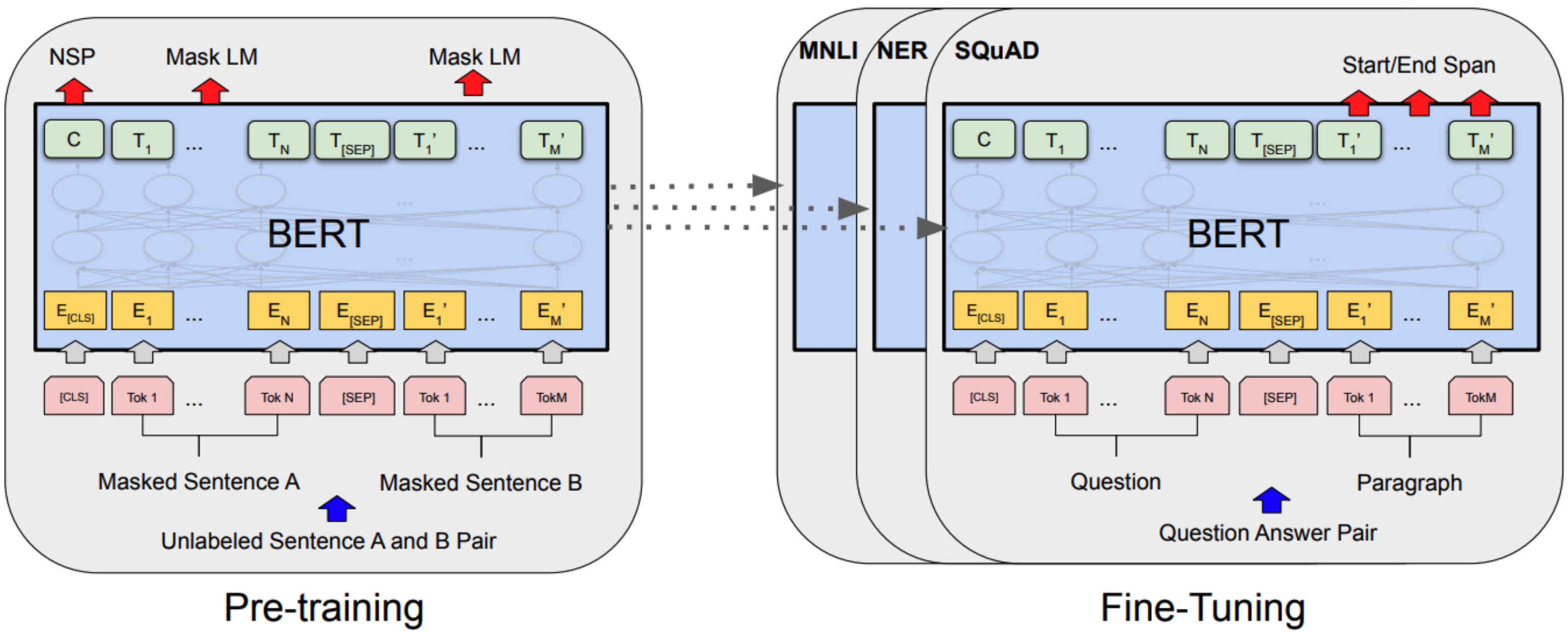
1. LSTM doesn't have the exploding/vanishing gradients issue as it occurs in RNNs
2. LSTM applies to sequential language tasks while RNNs applies to non-sequential language tasks
3. LSTM is better than RNN in most language tasks
4. LSTMs can be used for machine translation tasks

LSTM with attention

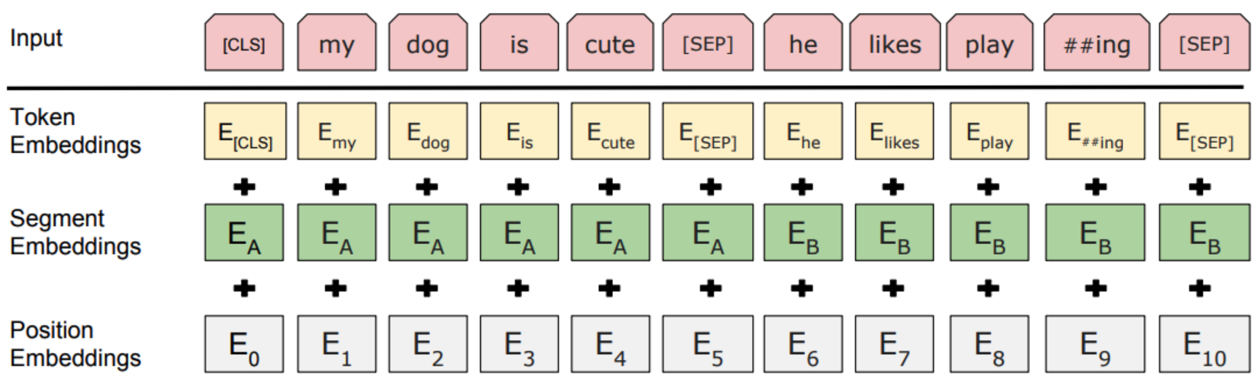


(a) Vanilla Encoder Decoder Architecture **(b) Attention Mechanism**

BERT - Bi-directional Encoders from Transformers



BERT Embeddings



BERT pre-training

Two Tasks

1. **Masked LM Model:** Mask a word in the middle of a sentence and have BERT predict the masked word
2. **Next-sentence prediction:** Predict the next sentence - Use both positive and negative labels. How are these generated?

BERT pre-training

Two Tasks

1. **Masked LM Model:** Mask a word in the middle of a sentence and have BERT predict the masked word
2. **Next-sentence prediction:** Predict the next sentence - Use both positive and negative labels. How are these generated?

ICE #4: Supervised or Un-supervised?

1. Are the above two tasks supervised or un-supervised?

BERT pre-training

Two Tasks

1. **Masked LM Model:** Mask a word in the middle of a sentence and have BERT predict the masked word
2. **Next-sentence prediction:** Predict the next sentence - Use both positive and negative labels. How are these generated?

ICE #4: Supervised or Un-supervised?

1. Are the above two tasks supervised or un-supervised?

Data set!

English Wikipedia and book corpus documents!

BERT - Bi-directional Encoders from Transformers

System	MNLI-(m/mm)	QQP	QNLI	SST-2	CoLA	STS-B	MRPC	RTE	**Average**
	392k	363k	108k	67k	8.5k	5.7k	3.5k	2.5k	-
Pre-OpenAI SOTA	80.6/80.1	66.1	82.3	93.2	35.0	81.0	86.0	61.7	74.0
BiLSTM+ELMo+Attn	76.4/76.1	64.8	79.8	90.4	36.0	73.3	84.9	56.8	71.0
OpenAI GPT	82.1/81.4	70.3	87.4	91.3	45.4	80.0	82.3	56.0	75.1
$\text{BERT}_{\text{BASE}}$	84.6/83.4	71.2	90.5	93.5	52.1	85.8	88.9	66.4	79.6
$\text{BERT}_{\text{LARGE}}$	**86.7/85.9**	**72.1**	**92.7**	**94.9**	**60.5**	**86.5**	**89.3**	**70.1**	**82.1**

BERT - Bi-directional Encoders from Transformers

System	Dev	Test
ESIM+GloVe	51.9	52.7
ESIM+ELMo	59.1	59.2
OpenAI GPT	-	78.0
$\text{BERT}_{\text{BASE}}$	81.6	-
$\text{BERT}_{\text{LARGE}}$	**86.6**	**86.3**
Human (expert)†	-	85.0
Human (5 annotations)†	-	88.0

Table 4: SWAG Dev and Test accuracies. †Human performance is measured with 100 samples, as reported in the SWAG paper.

ICE #6

MLM

What's the real point of using masked language models (MLM) as compared to regular language models (LM). Select ones that apply!

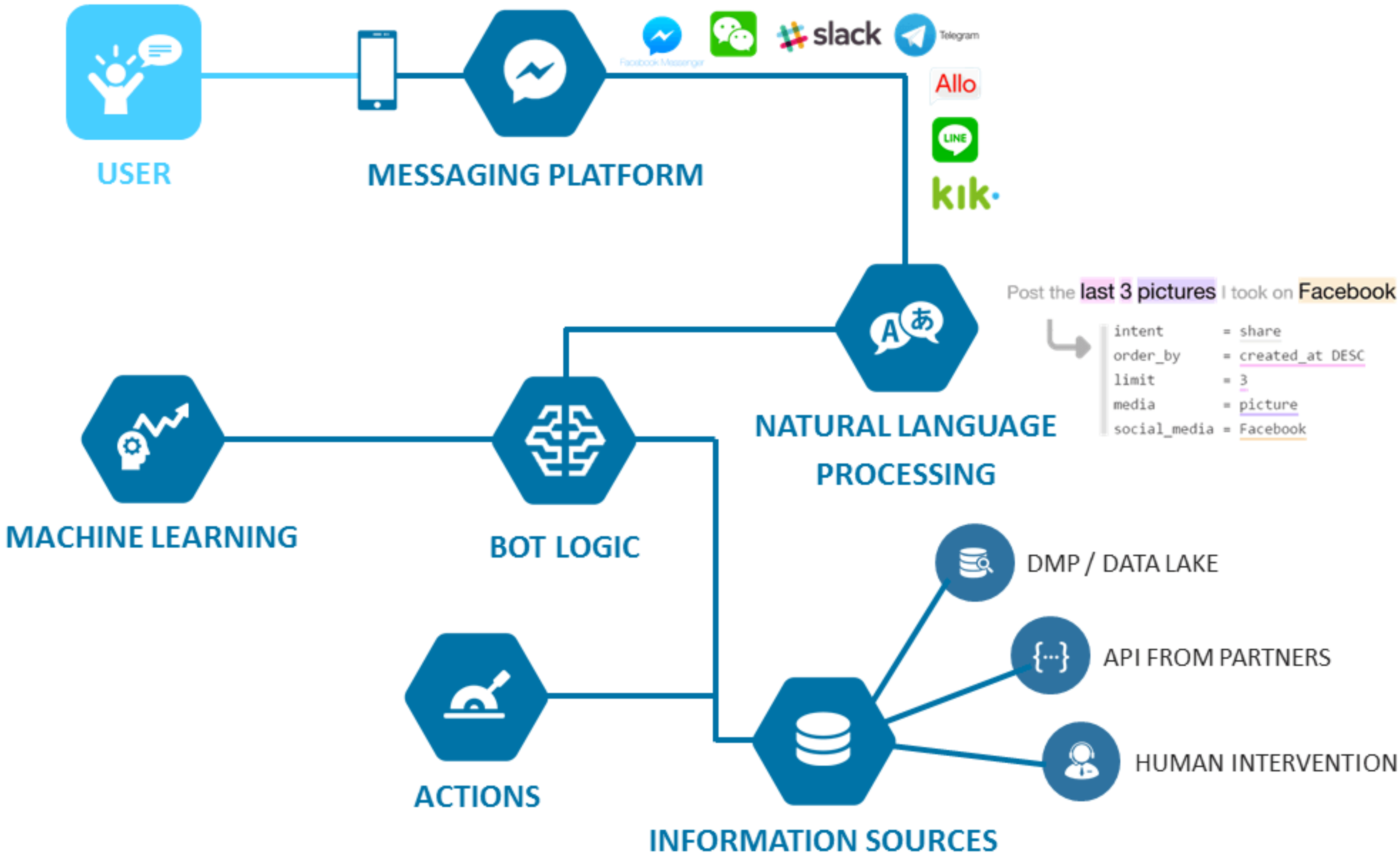
1. MLMs are used to learn how words fit together in a sentence
2. MLMs incorporate context from both directions and hence lead to better embeddings and predictions as compared to LMs
3. MLMs are great for complicated language tasks such as QA where you need to understand the sentence as a whole to give an appropriate answer to a question

Breakouts Time #1

Auto-complete — 5 mins

Let's say you are tasked with building an in-email auto-completion application, which can help complete partial sentences into full sentences through suggestions (auto-complete). How would you use what we have learned so far to model this? What architecture would you use? What would be your data? And what are some pitfalls or pain-points your model should address?

Chat Bots


USER
MESSAGING PLATFORM
slack
Telegram
Allo
kik
Post the last 3 pictures I took on Facebook
intent = share
order_by = created_at DESC
limit = 3
media = picture
social_media = Facebook
NATURAL LANGUAGE PROCESSING
MACHINE LEARNING
BOT LOGIC
DMP / DATA LAKE
API FROM PARTNERS
HUMAN INTERVENTION
ACTIONS
INFORMATION SOURCES

Breakouts Time #2

Retrieving Tables with Chat bots — 7 mins

You are building a chat-bot product at your company where queries come in from customers that own data in your company's cloud service. Your chat-bot responds retrieves the right table or combination of tables (through merge/filter operations) that contains this information or returns back with follow up questions to get more precise information or get back with a "Sorry, I don't have that information" response. How would you go about building a chat-bot like this? What data would you use? What ML models would you use, would it be supervised or un-supervised learning? What would be your evaluation metric? How would you test if your chat bot is accurate in its responses?