DSC190: Machine Learning with Few Labels

Variational Autoencoders
Self-supervised Learning

Zhiting Hu

Lecture 5, October 7, 2021

UC San Diego
HALICIOĞLU DATA SCIENCE INSTITUTE

Logistics

- Homework 1 released today

Outline

- Variational inference (cont'd)
  - Variational autoencoders (VAEs)

- Self-supervised learning

Variational Auto-Encoders (VAEs)

VAEs are a combination of the following ideas:

- Variational Inference
  - ELBO
- Variational distribution parametrized as neural networks
- Reparameterization trick

[Courtesy: Dhruv, CS 4803]

Variational Auto-Encoders (VAEs)

- Model $p_\theta(\boldsymbol{x}, \boldsymbol{z}) = p_\theta(\boldsymbol{x}|\boldsymbol{z})p(\boldsymbol{z})$
  - $p_\theta(\boldsymbol{x}|\boldsymbol{z})$: a.k.a., generative model, generator, (probabilistic) decoder, …
  - $p(\boldsymbol{z})$: prior, e.g., Gaussian
- Assume variational distribution $q_\phi(\boldsymbol{z}|\boldsymbol{x})$
  - E.g., a Gaussian distribution parameterized as **deep neural networks**
  - a.k.a, recognition model, inference network, (probabilistic) encoder, …
- ELBO:

$$\mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\phi}; \boldsymbol{x}) = \mathrm{E}_{q_\phi(\boldsymbol{z}|\boldsymbol{x})}[\log p_\theta(\boldsymbol{x}, \boldsymbol{z})] + \mathrm{H}(q_\phi(\boldsymbol{z}|\boldsymbol{x}))$$
$$= \mathrm{E}_{q_\phi(\boldsymbol{z}|\boldsymbol{x})}[\log p_\theta(\boldsymbol{x}|\boldsymbol{z})] - \mathrm{KL}(q_\phi(\boldsymbol{z}|\boldsymbol{x}) \,||\, p(\boldsymbol{z}))$$

Reconstruction

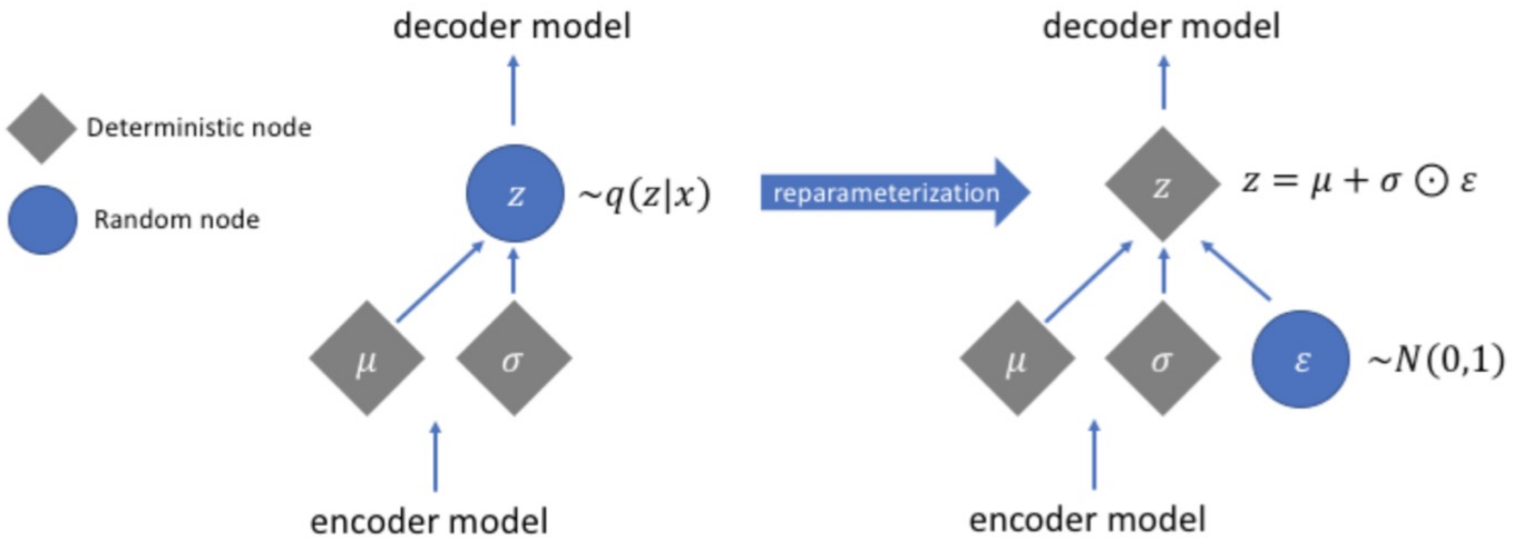
Divergence from prior
(KL divergence between two Guassians
has an analytic form)

Variational Auto-Encoders (VAEs)

- ELBO:

$$\mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\phi}; \boldsymbol{x}) = \mathrm{E}_{q_\phi(\boldsymbol{z}|\boldsymbol{x})}[\log p_\theta(\boldsymbol{x}, \boldsymbol{z})] + \mathrm{H}(q_\phi(\boldsymbol{z}|\boldsymbol{x}))$$
$$= \mathrm{E}_{q_\phi(\boldsymbol{z}|\boldsymbol{x})}[\log p_\theta(\boldsymbol{x}|\boldsymbol{z})] - \mathrm{KL}(q_\phi(\boldsymbol{z}|\boldsymbol{x}) \,||\, p(\boldsymbol{z}))$$

- Reparameterization:
  - $[\boldsymbol{\mu}; \boldsymbol{\sigma}] = f_\phi(\boldsymbol{x})$ (a neural network)
  - $\boldsymbol{z} = \boldsymbol{\mu} + \boldsymbol{\sigma} \odot \boldsymbol{\epsilon}, \quad \boldsymbol{\epsilon} \sim \boldsymbol{N}(\boldsymbol{0}, \boldsymbol{1})$



Variational Auto-Encoders (VAEs)

- ELBO:

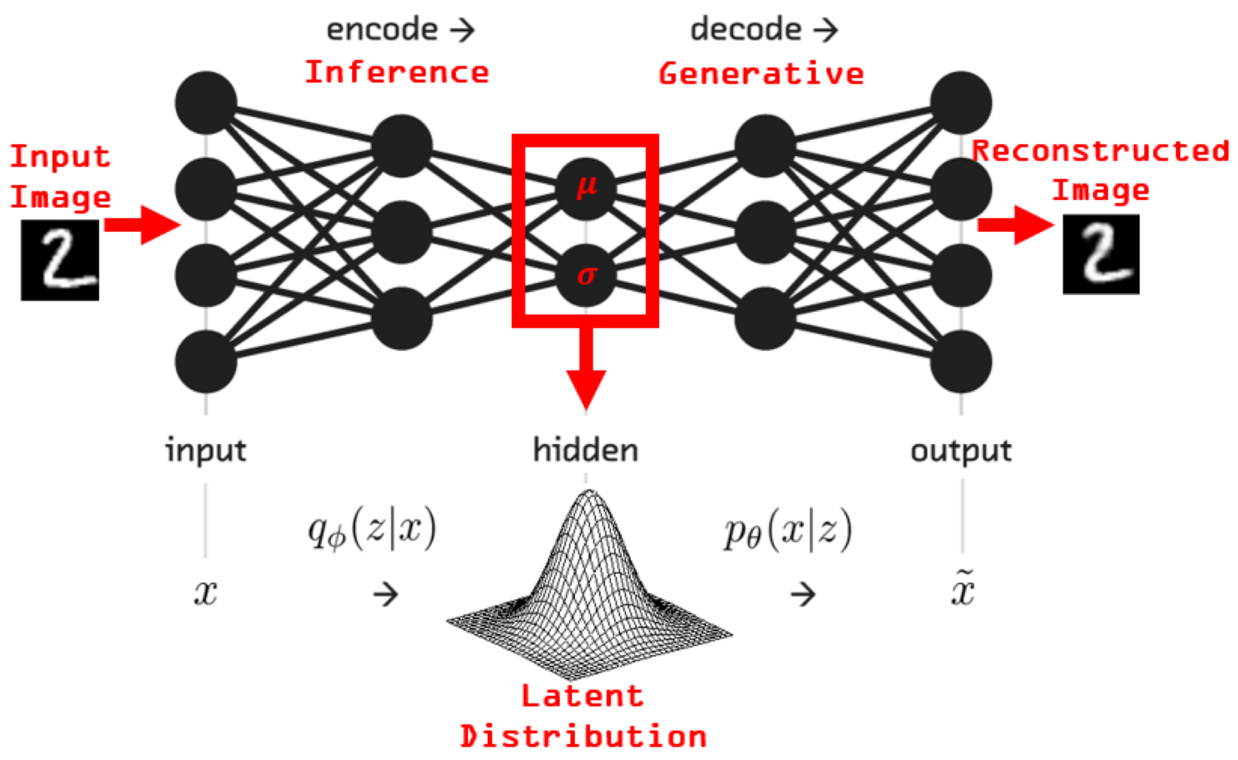
$$\mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\phi}; \boldsymbol{x}) = \mathrm{E}_{q_\phi(\boldsymbol{z}|\boldsymbol{x})}[\log p_\theta(\boldsymbol{x}, \boldsymbol{z})] + \mathrm{H}(q_\phi(\boldsymbol{z}|\boldsymbol{x}))$$
$$= \mathrm{E}_{q_\phi(\boldsymbol{z}|\boldsymbol{x})}[\log p_\theta(\boldsymbol{x}|\boldsymbol{z})] - \mathrm{KL}(q_\phi(\boldsymbol{z}|\boldsymbol{x}) \,||\, p(\boldsymbol{z}))$$

- Reparameterization:
  - $[\boldsymbol{\mu}; \boldsymbol{\sigma}] = f_\phi(\boldsymbol{x})$ (a neural network)
  - $\boldsymbol{z} = \boldsymbol{\mu} + \boldsymbol{\sigma} \odot \boldsymbol{\epsilon}, \quad \boldsymbol{\epsilon} \sim \boldsymbol{N}(\boldsymbol{0}, \boldsymbol{1})$

$$\nabla_{\boldsymbol{\phi}} \mathcal{L} = \mathrm{E}_{\epsilon \sim N(\boldsymbol{0},\boldsymbol{1})}[\, \nabla_{\boldsymbol{z}}[\log p_\theta(\boldsymbol{x}, \boldsymbol{z}) - \log q_\phi(\boldsymbol{z}|\boldsymbol{x})] \, \nabla_\phi z(\epsilon, \boldsymbol{\phi})]$$

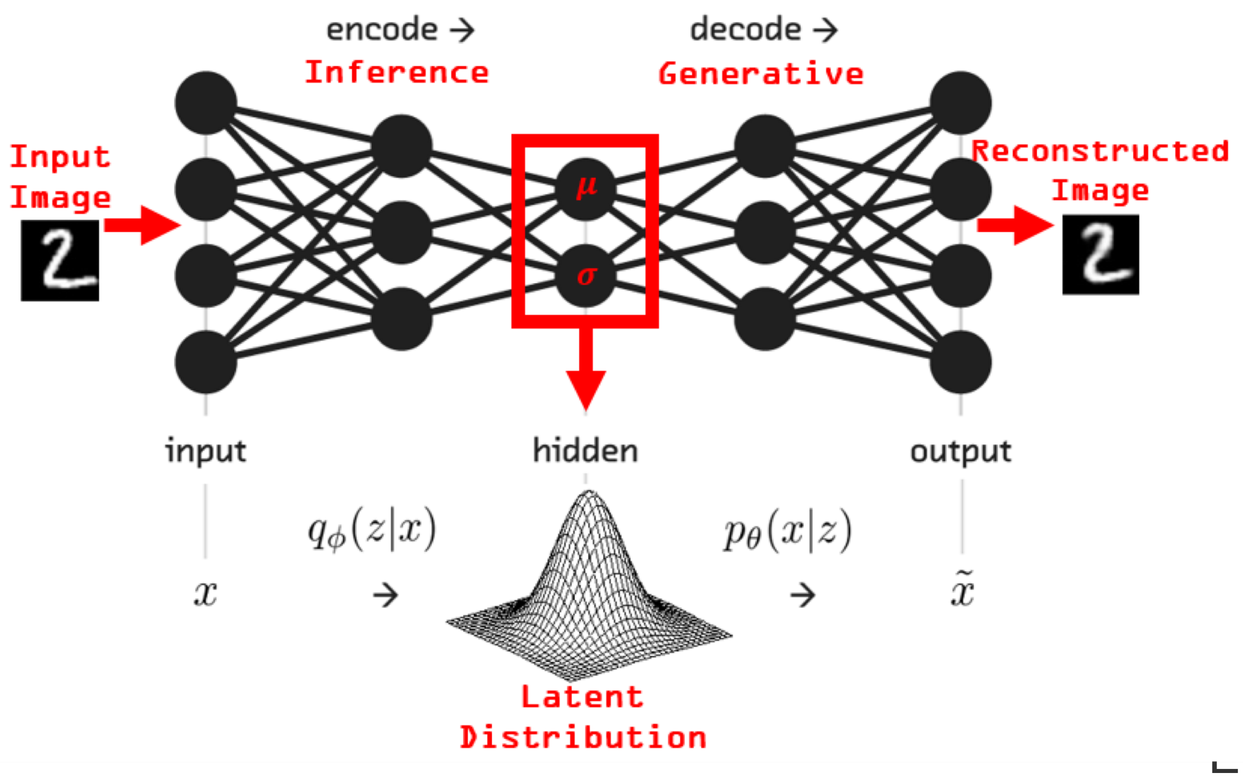
$$\nabla_\theta \mathcal{L} = \mathrm{E}_{q_\phi(\boldsymbol{z}|\boldsymbol{x})}[\nabla_\theta \log p_\theta(\boldsymbol{x}, \boldsymbol{z})] - \mathrm{H}(q_\phi(\boldsymbol{z}|\boldsymbol{x}))$$

Example: VAEs for images



[https://www.kaggle.com/rvislaywade/visualizing-mnist-using-a-variational-autoencoder]

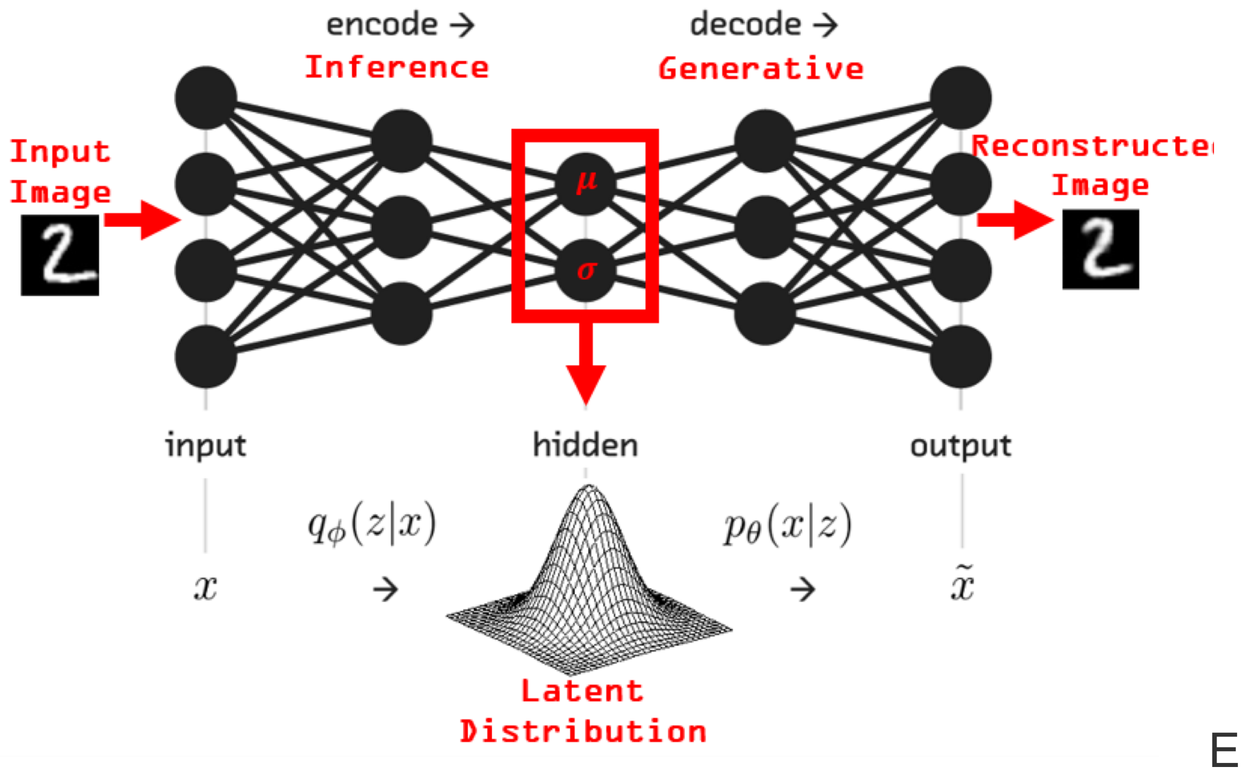
Example: VAEs for images

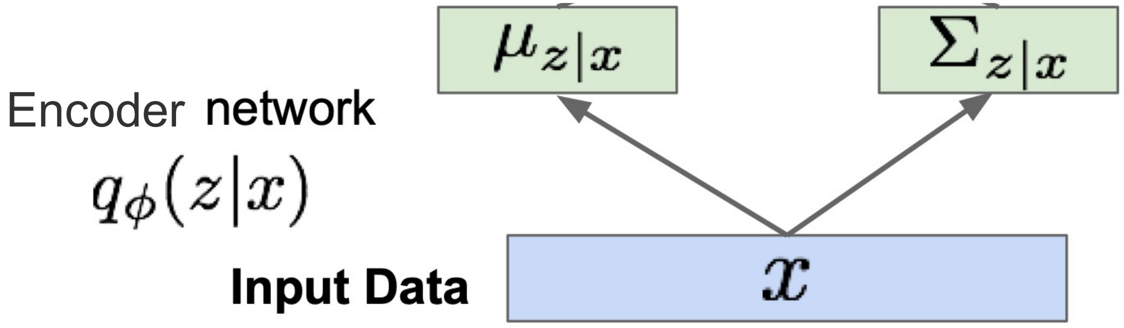


Input Data x

[Courtesy: Fei-Fei Li, Justin Johnson, Serena Yeung, CS 231n]

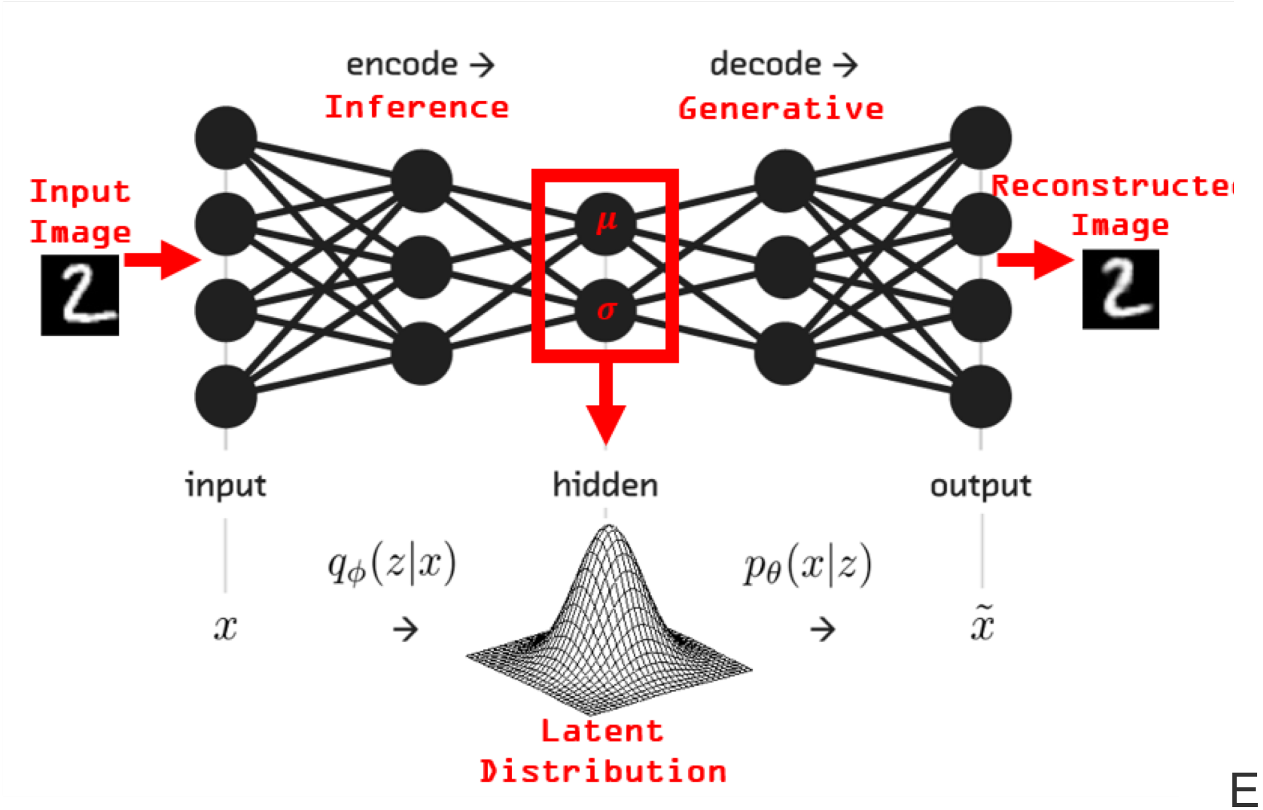
Example: VAEs for images

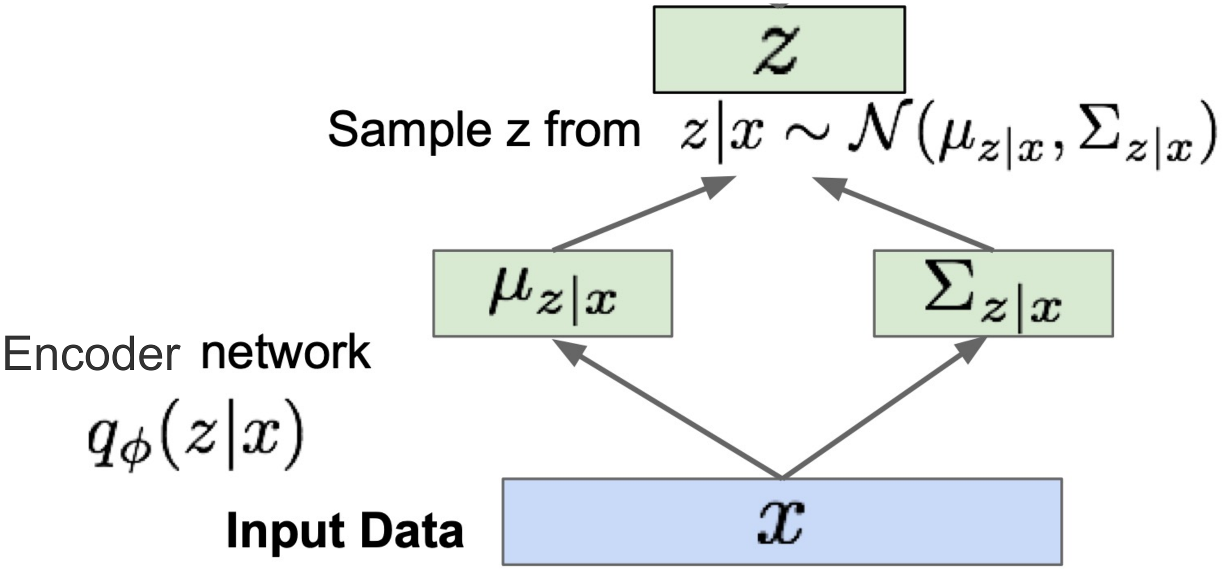




[Courtesy: Fei-Fei Li, Justin Johnson, Serena Yeung, CS 231n]

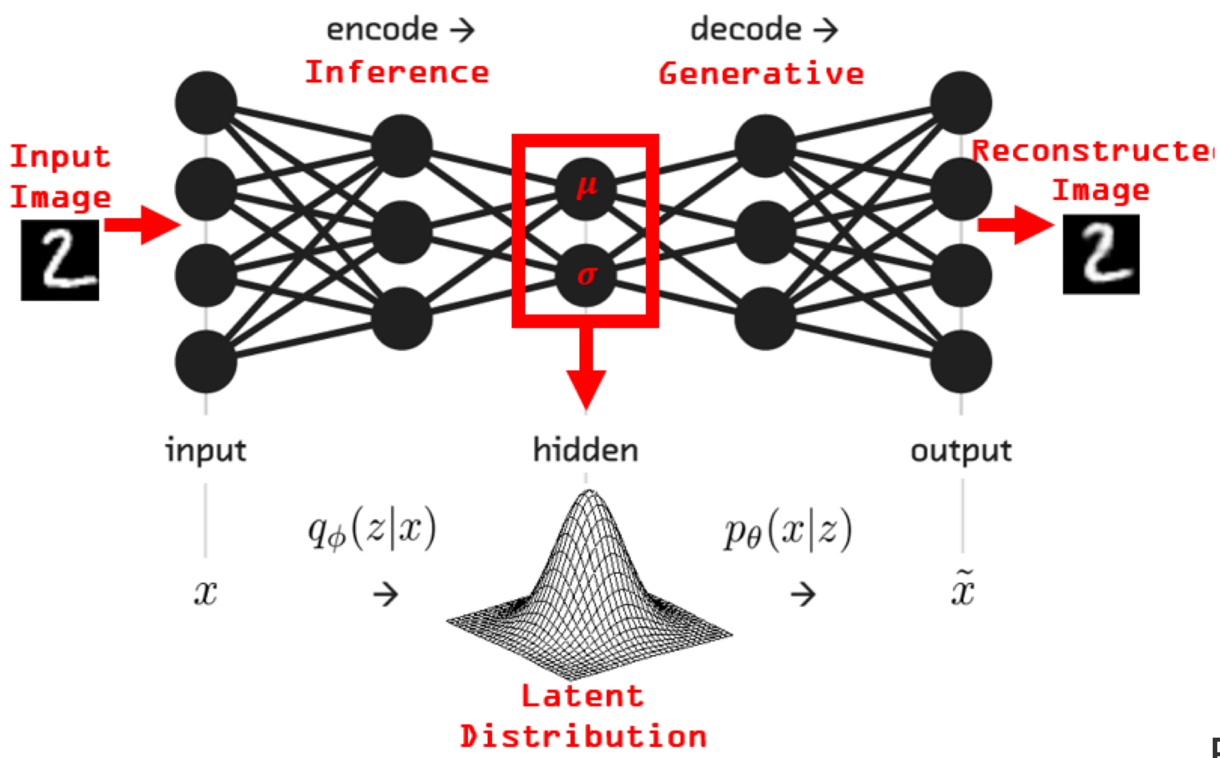
Example: VAEs for images

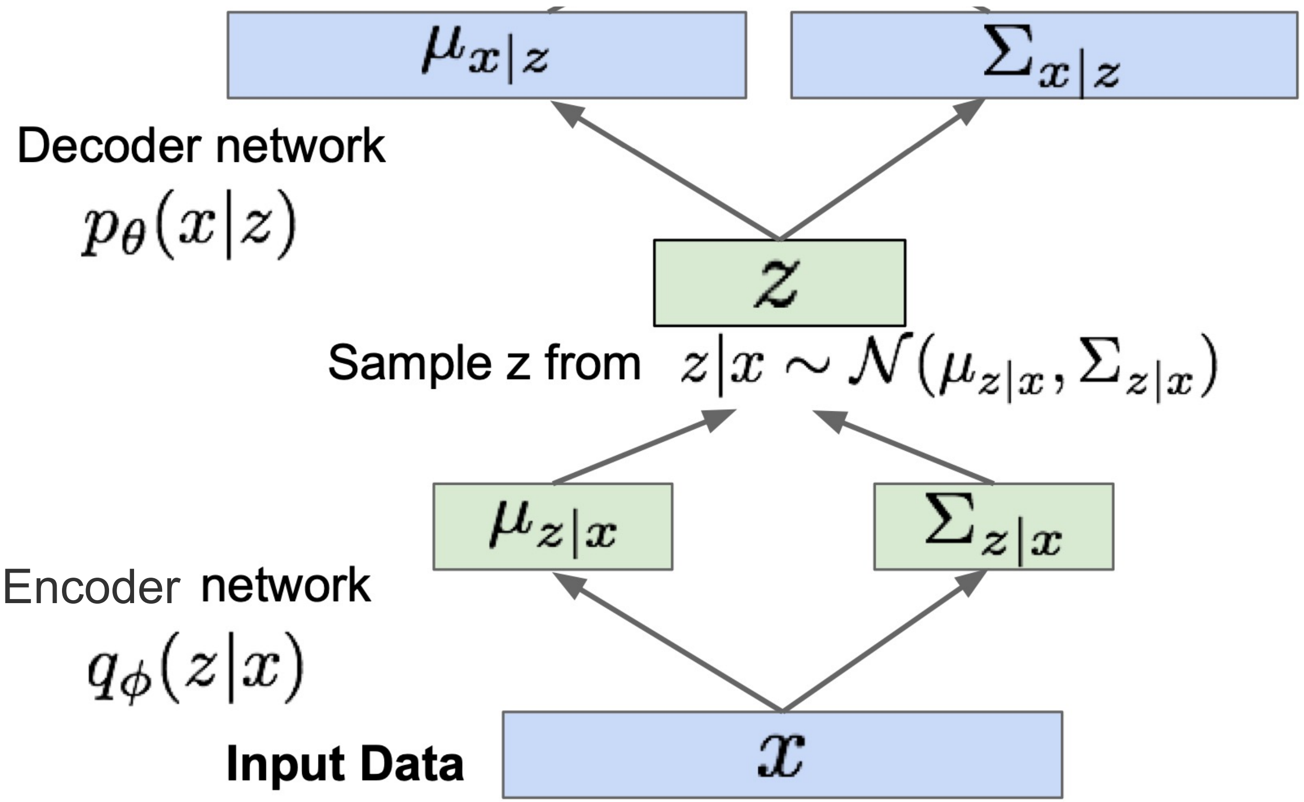




[Courtesy: Fei-Fei Li, Justin Johnson, Serena Yeung, CS 231n]

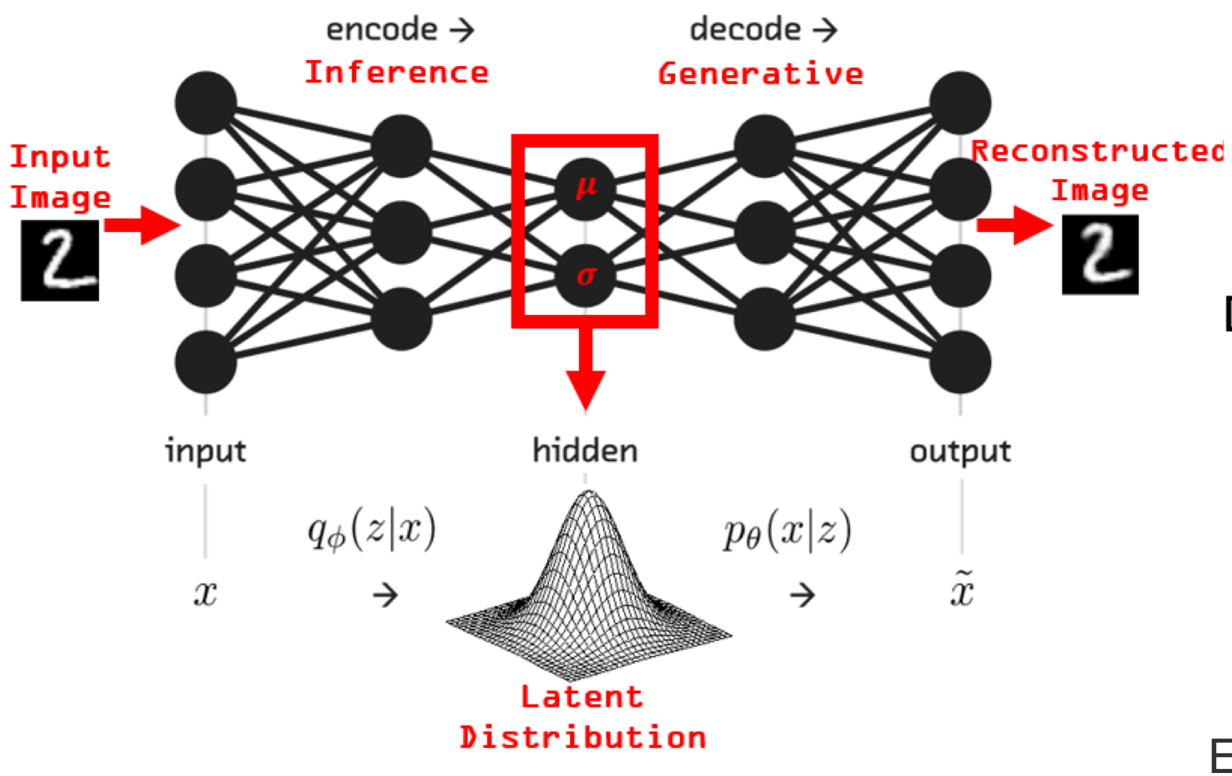
Example: VAEs for images

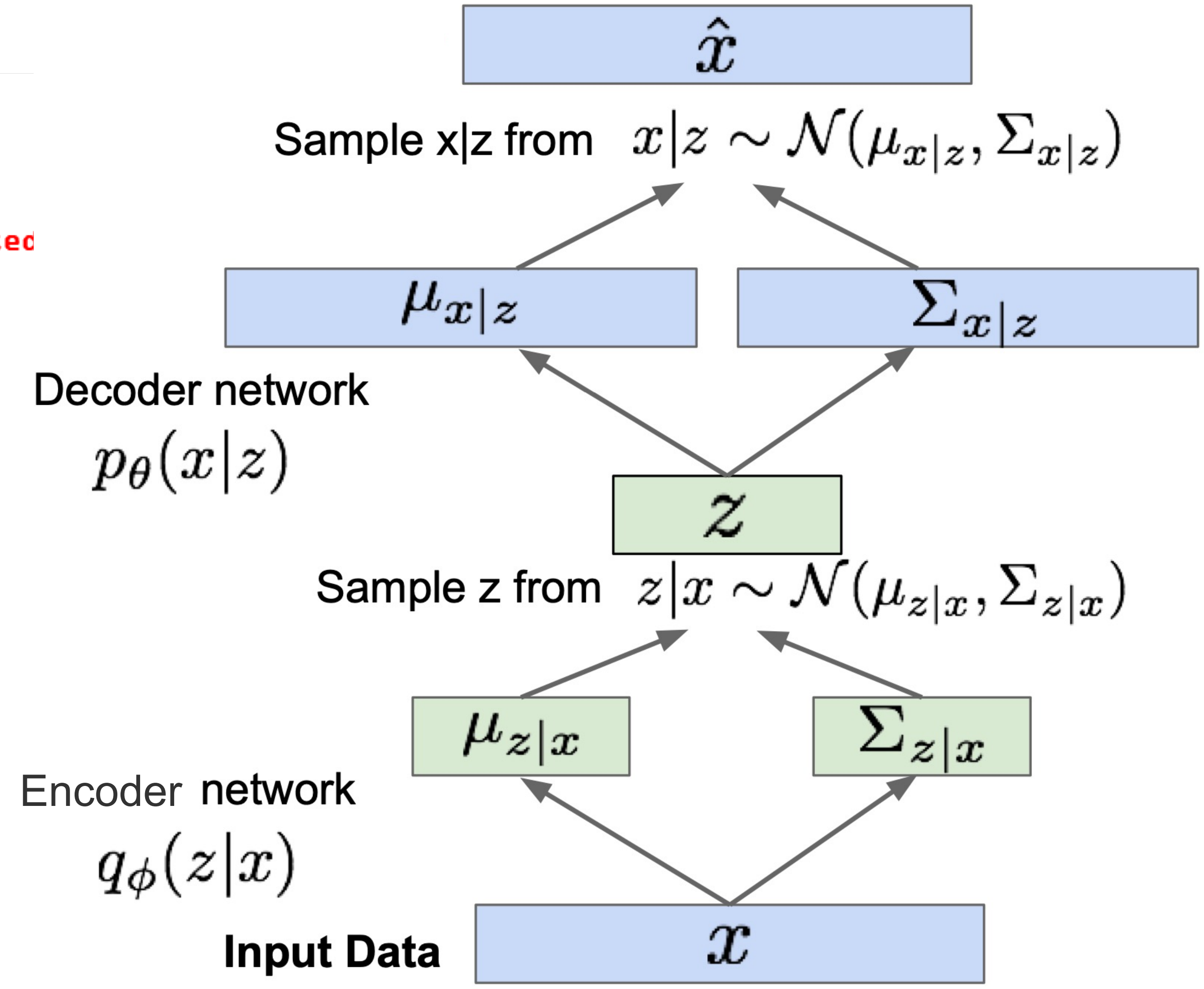




[Courtesy: Fei-Fei Li, Justin Johnson, Serena Yeung, CS 231n]

Example: VAEs for images



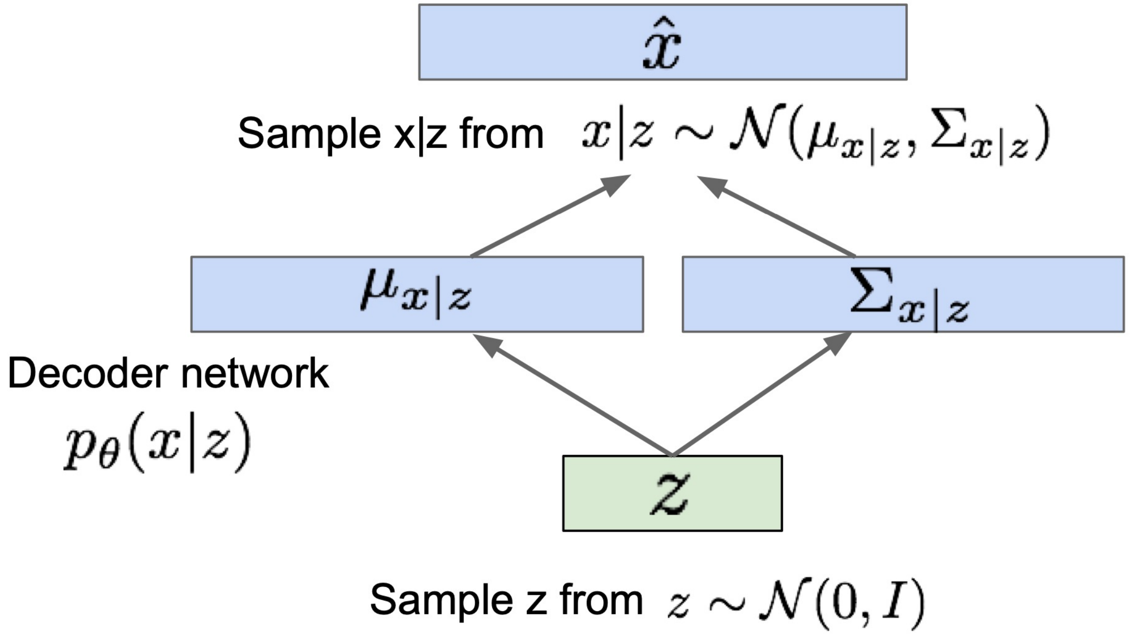


[Courtesy: Fei-Fei Li, Justin Johnson, Serena Yeung, CS 231n]

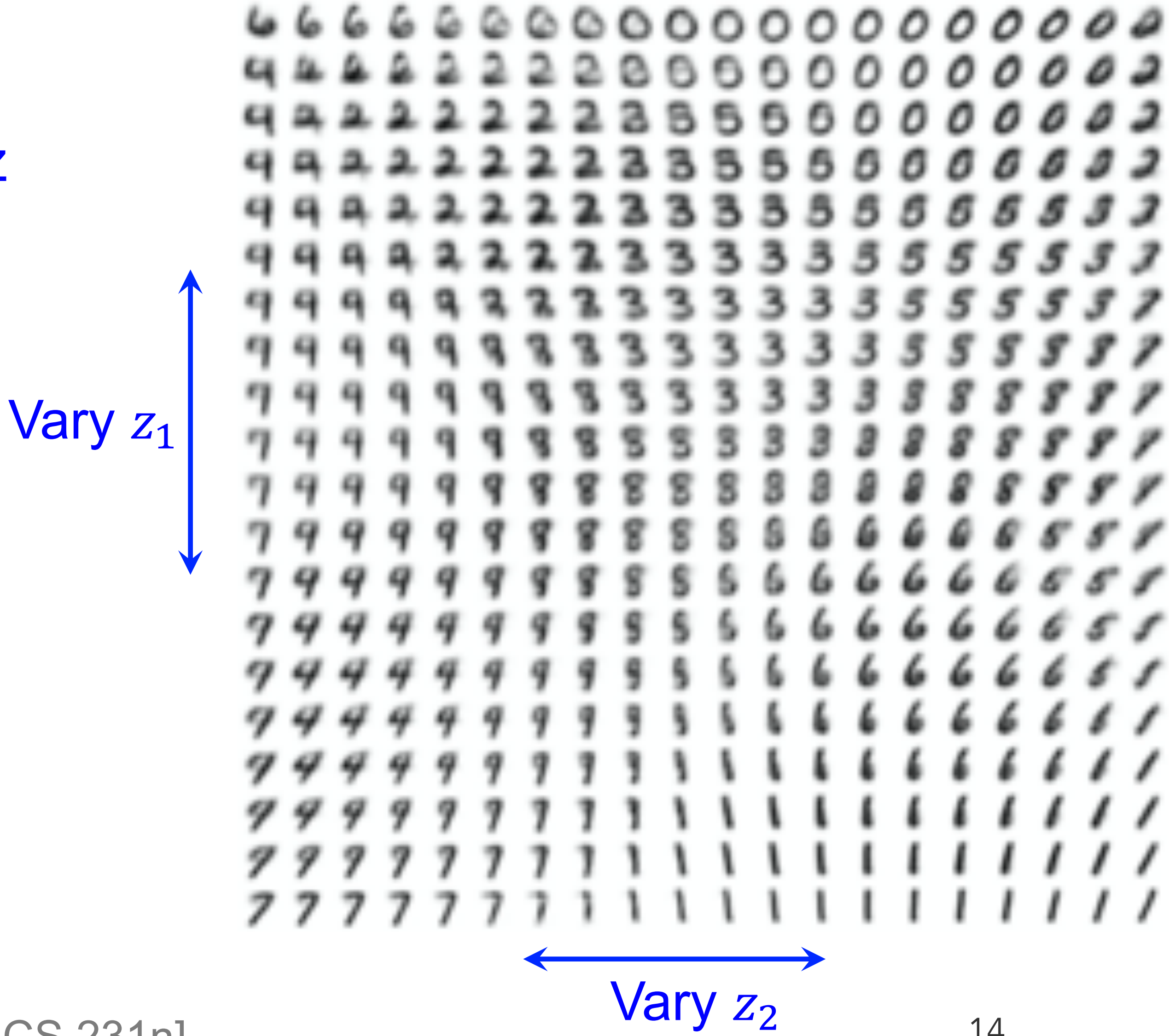
Example: VAEs for images

Generating samples:

- Use decoder network. Now sample z from prior!



Data manifold for 2-d **z**

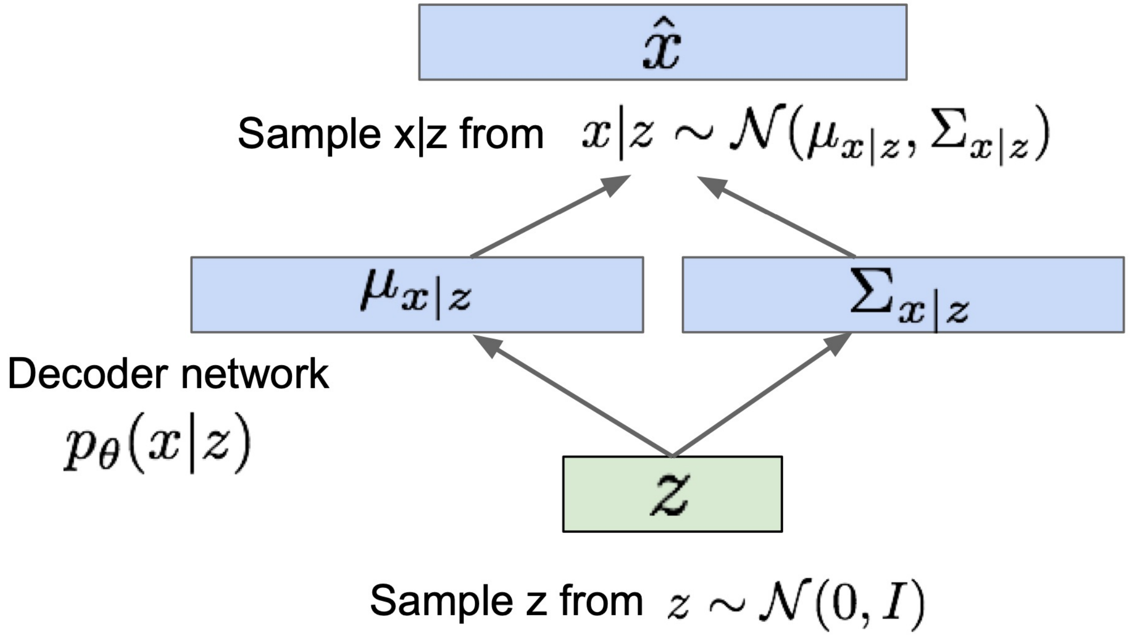


[Courtesy: Fei-Fei Li, Justin Johnson, Serena Yeung, CS 231n]

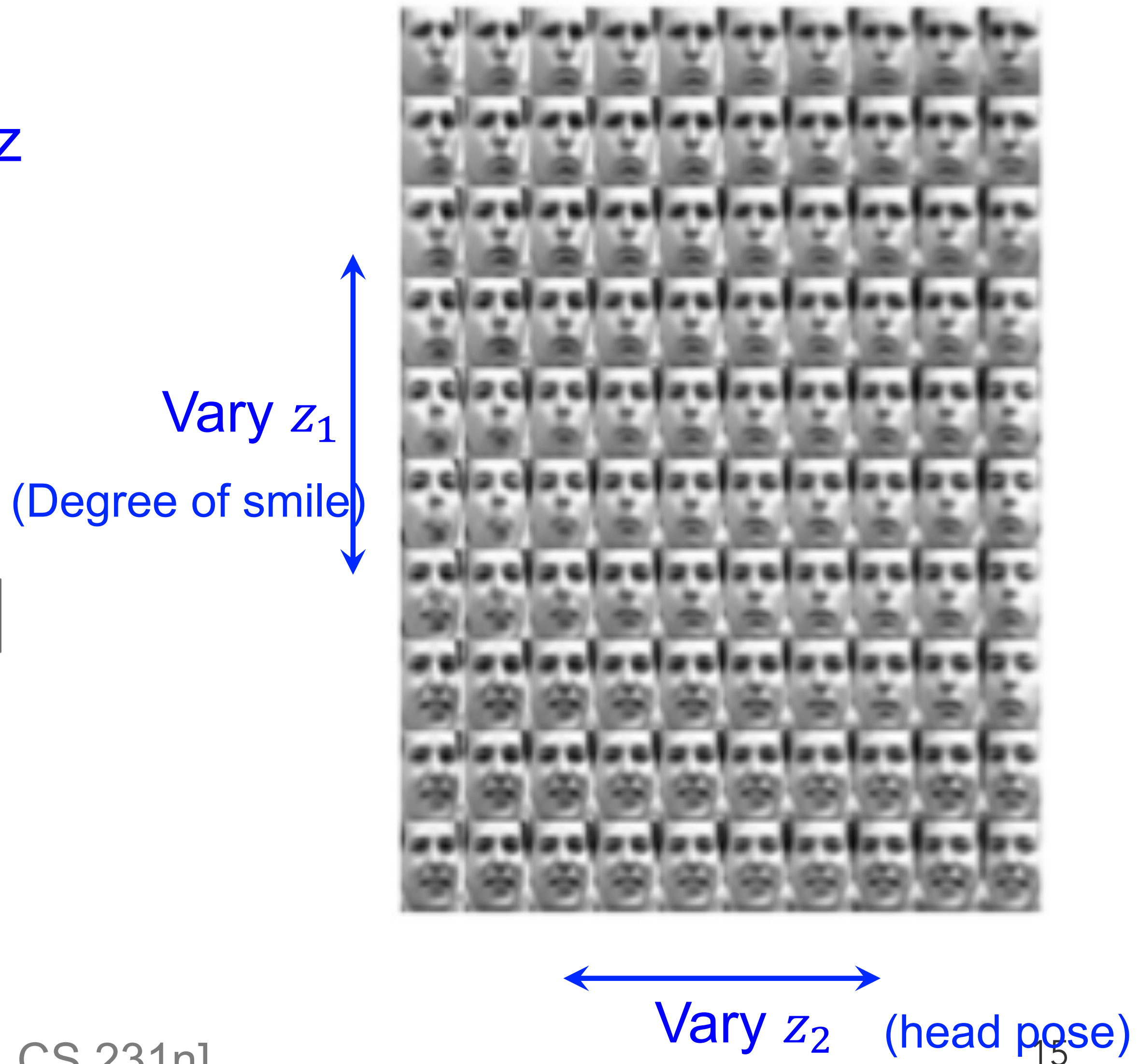
Example: VAEs for images

Generating samples:

- Use decoder network. Now sample z from prior!



Data manifold for 2-d **z**



[Courtesy: Fei-Fei Li, Justin Johnson, Serena Yeung, CS 231n]

Example: VAEs for text

- Latent code interpolation and sentences generation from VAEs [Bowman et al., 2015].

“ i want to talk to you . ”
“i want to be with you . ”
“i do n’t want to be with you . ”
i do n’t want to be with you .
she did n’t want to be with him .

Variational Auto-Encoders (VAEs)

Algorithm 1 Minibatch version of the Auto-Encoding VB (AEVB) algorithm. Either of the two SGVB estimators in section 2.3 can be used. We use settings $M = 100$ and $L = 1$ in experiments.

$\boldsymbol{\theta}, \boldsymbol{\phi} \leftarrow$ Initialize parameters
repeat
  $\mathbf{X}^M \leftarrow$ Random minibatch of M datapoints (drawn from full dataset)
  $\boldsymbol{\epsilon} \leftarrow$ Random samples from noise distribution $p(\boldsymbol{\epsilon})$
  $\mathbf{g} \leftarrow \nabla_{\boldsymbol{\theta},\boldsymbol{\phi}} \widetilde{\mathcal{L}}^M(\boldsymbol{\theta}, \boldsymbol{\phi}; \mathbf{X}^M, \boldsymbol{\epsilon})$ (Gradients of minibatch estimator (8))
  $\boldsymbol{\theta}, \boldsymbol{\phi} \leftarrow$ Update parameters using gradients $\mathbf{g}$ (e.g. SGD or Adagrad [DHS10])
until convergence of parameters $(\boldsymbol{\theta}, \boldsymbol{\phi})$
return $\boldsymbol{\theta}, \boldsymbol{\phi}$

[Kingma & Welling, 2014]

Note: Amortized Variational Inference

- Variational distribution as an inference model $q_{\phi}(\boldsymbol{z}|\boldsymbol{x})$ with parameters $\boldsymbol{\phi}$ (which was traditionally factored over samples)
- Amortize the cost of inference by learning a **single** data-dependent inference model
- The trained inference model can be used for quick inference on new data

Variational Auto-encoders: Summary

- A combination of the following ideas:
  - Variational Inference: ELBO
  - Variational distribution parametrized as neural networks
  - Reparameterization trick

$$\mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\phi}; \boldsymbol{x}) = [\log p_\theta(\boldsymbol{x}|\boldsymbol{z})] - \mathrm{KL}(q_\phi(\boldsymbol{z}|\boldsymbol{x}) \,||\, p(\boldsymbol{z}))$$

Reconstruction — Divergence from prior

(Razavi et al., 2019)

- Pros:
  - Principled approach to generative models
  - Allows inference of $q(z|x)$, can be useful feature representation for other tasks
- Cons:
  - Samples blurrier and lower quality compared to GANs
  - Tend to collapse on text data

Key Takeaways

- Stochastic VI
- Computing Gradients of Expectations $\mathcal{L} = \mathbb{E}_{q_{\lambda}(\boldsymbol{z})}[f_\lambda(\boldsymbol{z})]$
  - **Score gradient**

$$\nabla_\lambda \mathcal{L} = \mathbb{E}_{q_{\lambda}(\boldsymbol{z})}[f_\lambda(\boldsymbol{z}) \nabla_\lambda \log q_\theta(\boldsymbol{z}) + \nabla_\lambda f_\lambda(\boldsymbol{z})]$$

  - **Reparameterization gradient**

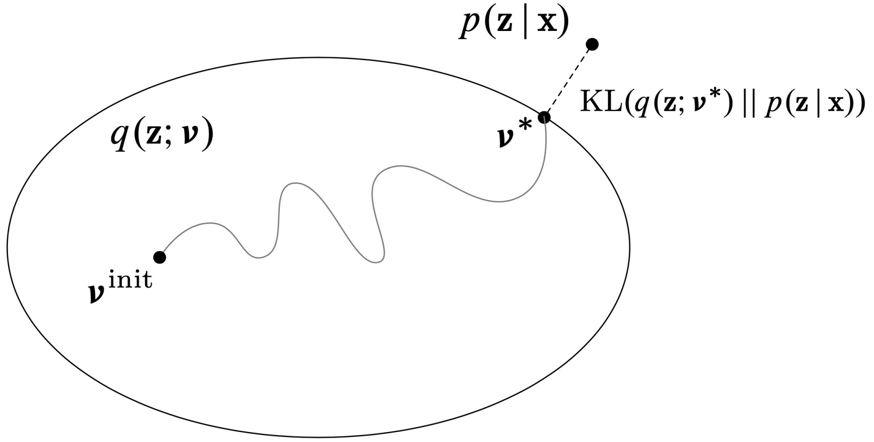
$$\nabla_\lambda \mathcal{L} = \mathbb{E}_{\boldsymbol{\epsilon} \sim \boldsymbol{s}(\boldsymbol{\epsilon})}[\nabla_{\boldsymbol{z}} f_\lambda(\boldsymbol{z}) \, \nabla_\lambda t(\epsilon, \lambda)]$$

- Black-box VI
- Variational autoencoders (VAEs)

Summary so far: Supervised Learning, Unsupervised Learning

- Supervised Learning
  - Maximum likelihood estimation (MLE)
  - Duality between MLE and Maximum Entropy Principle

- Unsupervised learning
  - Maximum likelihood estimation (MLE) with latent variables
    - Marginal log-likelihood
  - EM algorithm for MLE
    - ELBO
  - Variational Inference
    - ELBO
    - Variational distributions



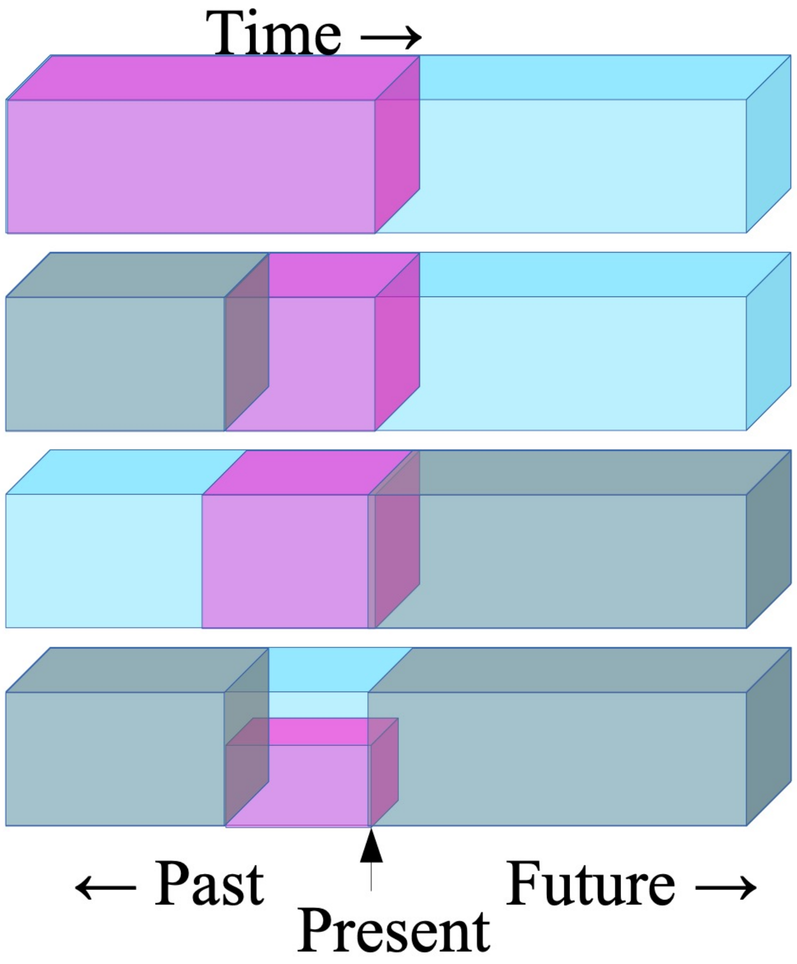
Self-Supervised Learning

Self-Supervised Learning

- Given an observed data instance $\boldsymbol{t}$
- One could derive various supervision signals based on the structure of the data
- By applying a "split" function that artificially partition $\boldsymbol{t}$ into two parts
  - $(\boldsymbol{x}, \boldsymbol{y}) = split(\boldsymbol{t})$
  - sometimes split in a stochastic way
- Treat $\boldsymbol{x}$ as the input and $\boldsymbol{y}$ as the output
- Train a model $p_\theta(\boldsymbol{y}|\boldsymbol{x})$

Self-Supervised Learning: Examples

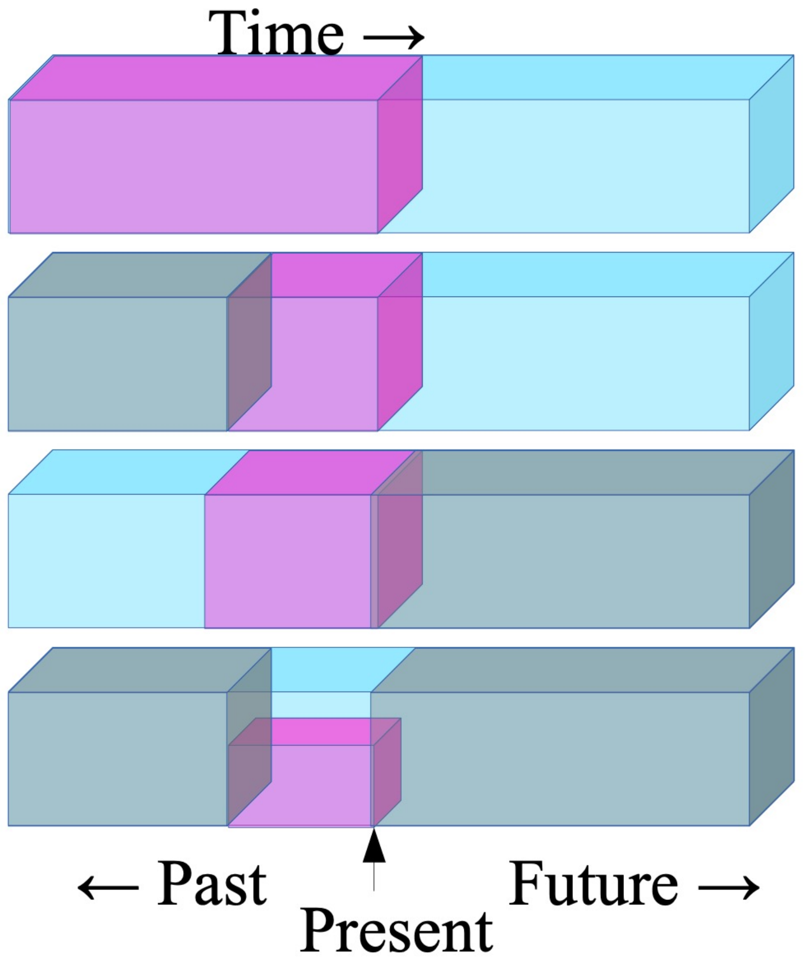
- **Predict any part of the input from any other part.**
- **Predict the future from the past.**
- **Predict the future from the recent past.**
- **Predict the past from the present.**
- **Predict the top from the bottom.**



[Courtesy: Lecun "Self-supervised Learning"]

Self-Supervised Learning: Examples

- **Predict any part of the input from any other part.**
- **Predict the future from the past.**
- **Predict the future from the recent past.**
- **Predict the past from the present.**
- **Predict the top from the bottom.**
- **Predict the occluded from the visible**
- **Pretend there is a part of the input you don't know and predict that.**



[Courtesy: Lecun "Self-supervised Learning"]

Self-Supervised Learning: Motivation (I)

- **Our brains do this all the time**

- **Filling in the visual field at the retinal blind spot**
- **Filling in occluded images, missing segments in speech**
- **Predicting the state of the world from partial (textual) descriptions**
- **Predicting the consequences of our actions**
- **Predicting the sequence of actions leading to a result**

- **Predicting any part of the past, present or future percepts from whatever information is available.**

[Courtesy: Lecun "Self-supervised Learning"]

Self-Supervised Learning: Motivation (I)

- Successfully learning to predict everything from everything else would result in the accumulation of lots of background knowledge about how the world works

- The model is forced to learn what we really care about, e.g. a semantic representation, in order to solve the prediction problem

[Courtesy: Lecun “Self-supervised Learning”]

[Courtesy: Zisserman “Self-supervised Learning”]

Self-Supervised Learning: Motivation (II)

- The machine predicts any part of its input from any observed part
  - **A lot of** supervision signals in each data instance

- Untapped/availability of vast numbers of unlabeled text/images/videos..
  - Facebook: one billion images uploaded per day
  - 300 hours of video are uploaded to YouTube every minute

[Courtesy: Zisserman “Self-supervised Learning”]

Self-Supervised Learning (SSL): Examples

- SSL from text
- SSL from images
- SSL from videos

Self-Supervised Learning from Text

Examples:

- Language models
- Learning contextual text representations

Language Models

- Calculates the probability of a sentence:
  - Sentence:

$$\boldsymbol{y} = (y_1, y_2, \ldots, y_T)$$

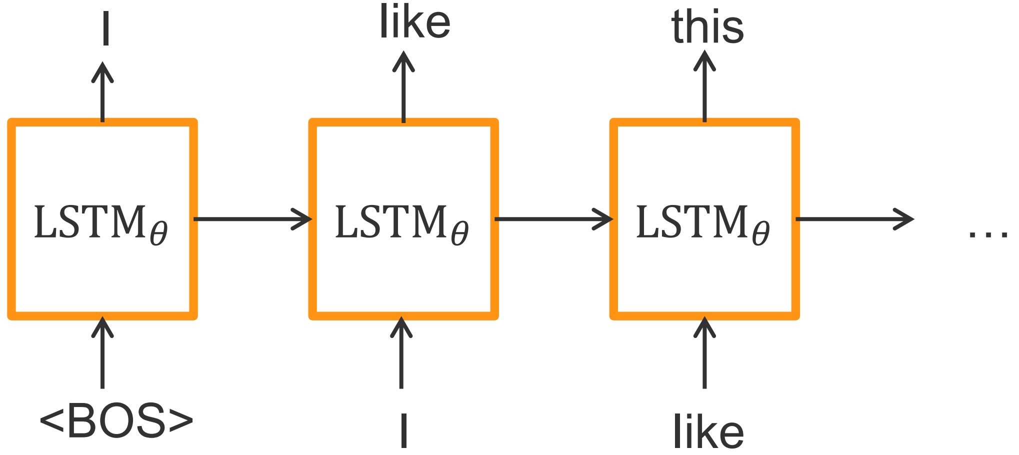
$$p_\theta(\boldsymbol{y}) = \prod_{t=1}^{T} p_\theta(y_t \mid \boldsymbol{y}_{1:t-1})$$

Example:

(I, like, this, …)

$$\cdots p_\theta\,(like \mid I)\, p_\theta(this \mid I, like) \cdots$$

Model: LSTM RNN



Language Models

- Calculates the probability of a sentence:
  - Sentence:

$$\boldsymbol{y} = (y_1, y_2, \dots, y_T)$$

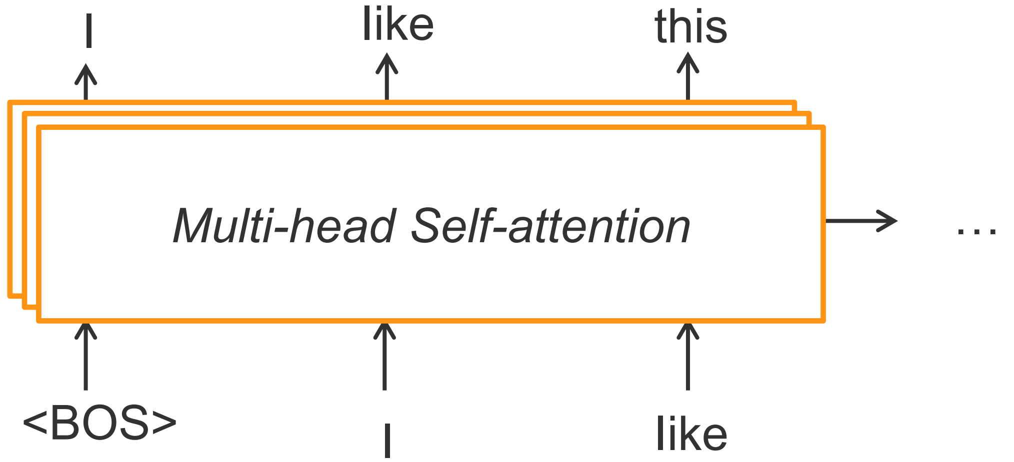
$$p_\theta(\boldsymbol{y}) = \prod_{t=1}^{T} p_\theta(y_t \mid \boldsymbol{y}_{1:t-1})$$

Example:

(I, like, this, …)

$$\cdots p_\theta\,(like \mid I)\, p_\theta(this \mid I, like) \cdots$$

Model: Transformer



Language Models: Training

- Given data example $\boldsymbol{y}^*$
- Minimizes negative log-likelihood of the data

$$\min_\theta \mathcal{L}_{\text{MLE}} = -\log p_\theta(\boldsymbol{y}^*) = -\prod_{t=1}^{T} p_\theta(y_t^* \mid \boldsymbol{y}_{1:t-1}^*)$$

  - Next word prediction
  - Inference: teacher-forcing decoding
    - For every step t, feed in the previous ground-truth tokens $y_{1:t-1}^*$ to decode next step

Language Models: GPT3

- A Transformer-based LM with 125M to 175B parameters
- Trained on massive text data

Dataset	# Tokens (Billions)
Total	499
Common Crawl (filtered by quality)	410
WebText2	19
Books1	12
Books2	55
Wikipedia	3

Brown et al., 2020 "Language Models Are Few-Shot Learners"

[Table from https://lambdalabs.com/blog/demystifying-gpt-3/]

Language Models: GPT3

- Generation

```
Title: United Methodists Agree to Historic Split
Subtitle: Those who oppose gay marriage will form their own denomination
Article: After two days of intense debate, the United Methodist Church
has agreed to a historic split - one that is expected to end in the
creation of a new denomination, one that will be "theologically and
socially conservative," according to The Washington Post. The majority of
delegates attending the church's annual General Conference in May voted to
strengthen a ban on the ordination of LGBTQ clergy and to write new rules
that will "discipline" clergy who officiate at same-sex weddings. But
those who opposed these measures have a new plan: They say they will form a
separate denomination by 2020, calling their church the Christian Methodist
denomination.
The Post notes that the denomination, which claims 12.5 million members, was
in the early 20th century the "largest Protestant denomination in the U.S.,"
but that it has been shrinking in recent decades. The new split will be the
second in the church's history. The first occurred in 1968, when roughly
10 percent of the denomination left to form the Evangelical United Brethren
Church. The Post notes that the proposed split "comes at a critical time
for the church, which has been losing members for years," which has been
"pushed toward the brink of a schism over the role of LGBTQ people in the
church." Gay marriage is not the only issue that has divided the church. In
2016, the denomination was split over ordination of transgender clergy, with
the North Pacific regional conference voting to ban them from serving as
clergy, and the South Pacific regional conference voting to allow them.
```

Figure 3.14: The GPT-3 generated news article that humans had the greatest difficulty distinguishing from a human written article (accuracy: 12%).

Language Models: GPT3

- Few-shot prediction

Prompts in Red. GPT3 responses in Blue

Q. How do you draw a bicycle?
A. You draw a bicycle with two circles connected by a line.

Q. How do you draw a ball?
A. You draw a ball with one circle.

Q. How do you draw a house?
A. You draw a house with a trapezium on top of a square.

Q. How do you draw a rabbit?
A. You draw a rabbit with four circles: one large circle for the head, a small one for the nose, a large circle for the body, and two tiny circles for the feet.

Q. How do you draw a snail?
A. You draw a snail with six circles: one large circle for the shell, two small circles for the antennae, one large circle for the head, one small circle for the tail, and a small circle for the mouth.

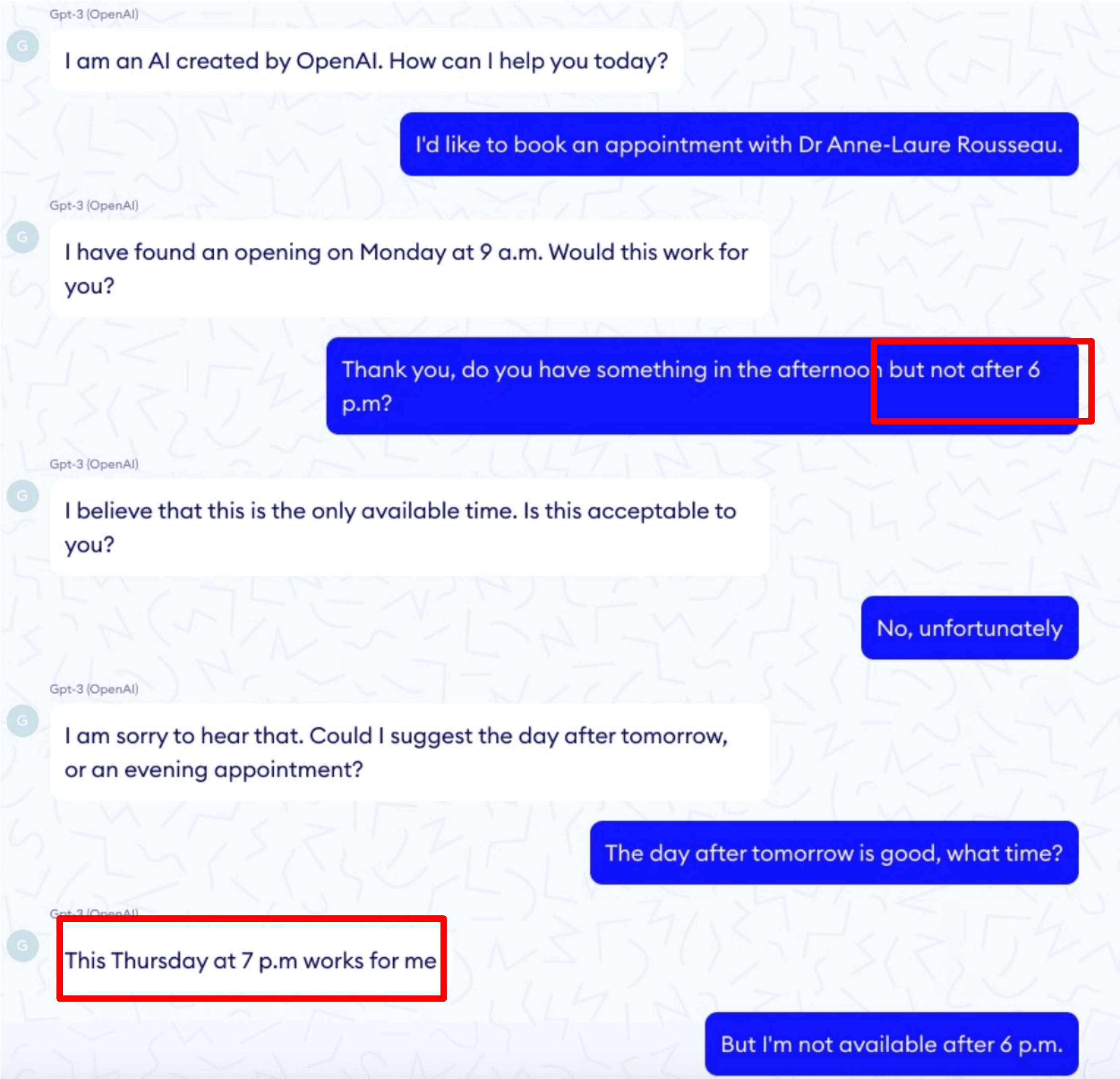
Q. How do you draw a monkey?
A. You draw a monkey with three circles: one large circle for the head, a small circle for the nose, and a medium-sized circle for the body.

Q. How do you draw a person?
A. You draw a person with three circles: one large circle for the head, a medium-sized circle for the body, and a small circle for the feet.

[Nurecas.com]

Language Models: GPT3

- Weakness



[Credit: Nabla]

Self-Supervised Learning from Text

Examples:

- Language models
- Learning contextual text representations

Word Embedding

- Conventional word embedding:
  - Word2vec, Glove
  - A pre-trained **matrix**, each row is an embedding vector of a word

	0	1	2	3	4	5	6	7	8	9	...
fox	-0.348680	-0.077720	0.177750	-0.094953	-0.452890	0.237790	0.209440	0.037886	0.035064	0.899010	...
ham	-0.773320	-0.282540	0.580760	0.841480	0.258540	0.585210	-0.021890	-0.463680	0.139070	0.658720	...
brown	-0.374120	-0.076264	0.109260	0.186620	0.029943	0.182700	-0.631980	0.133060	-0.128980	0.603430	...
beautiful	0.171200	0.534390	-0.348540	-0.097234	0.101800	-0.170860	0.295650	-0.041816	-0.516550	2.117200	...
jumps	-0.334840	0.215990	-0.350440	-0.260020	0.411070	0.154010	-0.386110	0.206380	0.386700	1.460500	...
eggs	-0.417810	-0.035192	-0.126150	-0.215930	-0.669740	0.513250	-0.797090	-0.068611	0.634660	1.256300	...
beans	-0.423290	-0.264500	0.200870	0.082187	0.066944	1.027600	-0.989140	-0.259950	0.145960	0.766450	...
sky	0.312550	-0.303080	0.019587	-0.354940	0.100180	-0.141530	-0.514270	0.886110	-0.530540	1.556600	...
bacon	-0.430730	-0.016025	0.484620	0.101390	-0.299200	0.761820	-0.353130	-0.325290	0.156730	0.873210	...
breakfast	0.073378	0.227670	0.208420	-0.456790	-0.078219	0.601960	-0.024494	-0.467980	0.054627	2.283700	...
toast	0.130740	-0.193730	0.253270	0.090102	-0.272580	-0.030571	0.096945	-0.115060	0.484000	0.848380	...
today	-0.156570	0.594890	-0.031445	-0.077586	0.278630	-0.509210	-0.066350	-0.081890	-0.047986	2.803600	...
blue	0.129450	0.036518	0.032298	-0.060034	0.399840	-0.103020	-0.507880	0.076630	-0.422920	0.815730	...
green	-0.072368	0.233200	0.137260	-0.156630	0.248440	0.349870	-0.241700	-0.091426	-0.530150	1.341300	...
kings	0.259230	-0.854690	0.360010	-0.642000	0.568530	-0.321420	0.173250	0.133030	-0.089720	1.528600	...
dog	-0.057120	0.052685	0.003026	-0.048517	0.007043	0.041856	-0.024704	-0.039783	0.009614	0.308416	...
sausages	-0.174290	-0.064869	-0.046976	0.287420	-0.128150	0.647630	0.056315	-0.240440	-0.025094	0.502220	...
lazy	-0.353320	-0.299710	-0.176230	-0.321940	-0.385640	0.586110	0.411160	-0.418680	0.073093	1.486500	...
love	0.139490	0.534530	-0.252470	-0.125650	0.048748	0.152440	0.199060	-0.065970	0.128830	2.055900	...
quick	-0.445630	0.191510	-0.249210	0.465900	0.161950	0.212780	-0.046480	0.021170	0.417660	1.686900	...

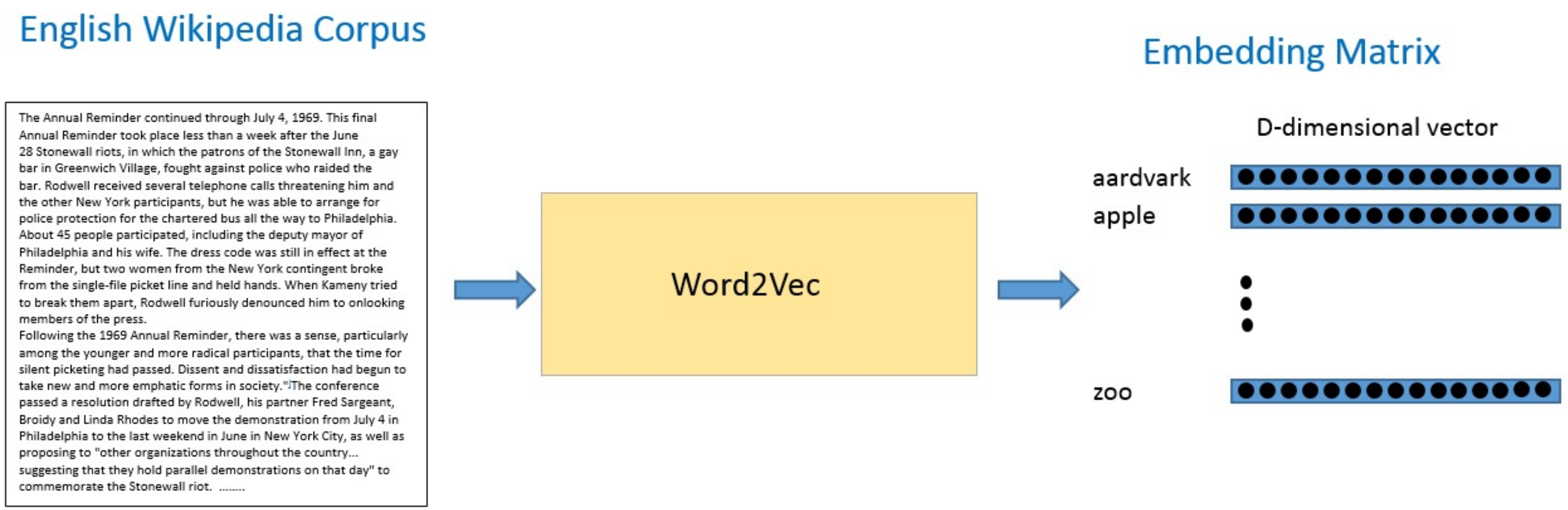
20 rows × 300 columns

[Courtesy: Vaswani, et al., 2017]

Word Embedding

- Conventional word embedding:
  - Word2vec, Glove
  - A pre-trained **matrix**, each row is an embedding vector of a word

	0	1	2	3	4	5	6	7	8	9	...
fox	-0.348680	-0.077720	0.177750	-0.094953	-0.452890	0.237790	0.209440	0.037886	0.035064	0.899010	...
ham	-0.773320	-0.282540	0.580760	0.841480	0.258540	0.585210	-0.021890	-0.463680	0.139070	0.658720	...
brown	-0.374120	-0.076264	0.109260	0.186620	0.029943	0.182700	-0.631980	0.133060	-0.128980	0.603430	...
beautiful	0.171200	0.534390	-0.348540	-0.097234	0.101800	-0.170860	0.295650	-0.041816	-0.516550	2.117200	...
jumps	-0.334840	0.215990	-0.350440	-0.260020	0.411070	0.154010	-0.386110	0.206380	0.386700	1.460500	...
eggs	-0.417810	-0.035192	-0.126150	-0.215930	-0.669740	0.513250	-0.797090	-0.068611	0.634660	1.256300	...
beans	-0.423290	-0.264500	0.200870	0.082187	0.066944	1.027600	-0.989140	-0.259950	0.145960	0.766450	...
sky	0.312550	-0.303080	0.019587	-0.354940	0.100180	-0.141530	-0.514270	0.886110	-0.530540	1.556600	...
bacon	-0.430730	-0.016025	0.484620	0.101390	-0.299200	0.761820	-0.353130	-0.325290	0.156730	0.873210	...
breakfast	0.073378	0.227670	0.208420	-0.456790	-0.078219	0.601960	-0.024494	-0.467980	0.054627	2.283700	...
							945	-0.115060	0.484000	0.848380	...
							350	-0.081890	-0.047986	2.803600	...
							880	0.076630	-0.422920	0.815730	...
							700	-0.091426	-0.530150	1.341300	...
							250	0.133030	-0.089720	1.528600	...
							704	-0.039783	0.009614	0.308416	...
							315	-0.240440	-0.025094	0.502220	...
							160	-0.418680	0.073093	1.486500	...
							060	-0.065970	0.128830	2.055900	...
							480	0.021170	0.417660	1.686900	...



[Image source: Va

Word Embedding

- **Problem**: word embeddings are applied in a context free manner

```
open a bank account          on the river bank

             [0.3, 0.2, -0.8, …]
```

Courtesy: Devlin 2019

Word Embedding

- **Problem**: word embeddings are applied in a context free manner

open a bank account   on the river bank

[0.3, 0.2, -0.8, …]

- **Solution**: Train contextual representations on text corpus

[0.9, -0.2, 1.6, …]   [-1.9, -0.4, 0.1, …]

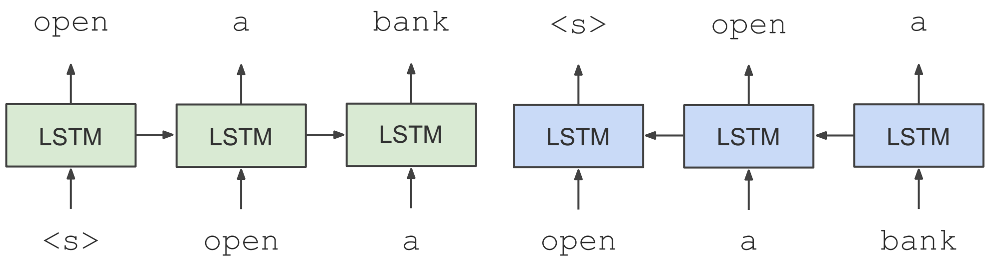
open a bank account   on the river bank

Courtesy: Devlin 2019

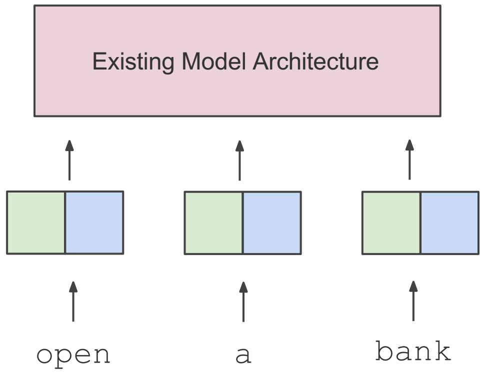
Contextual Representations

- *ELMo: Deep Contextual Word Embeddings*, AI2 & University of Washington, 2017

Train Separate Left-to-Right and Right-to-Left LMs



Apply as "Pre-trained Embeddings"



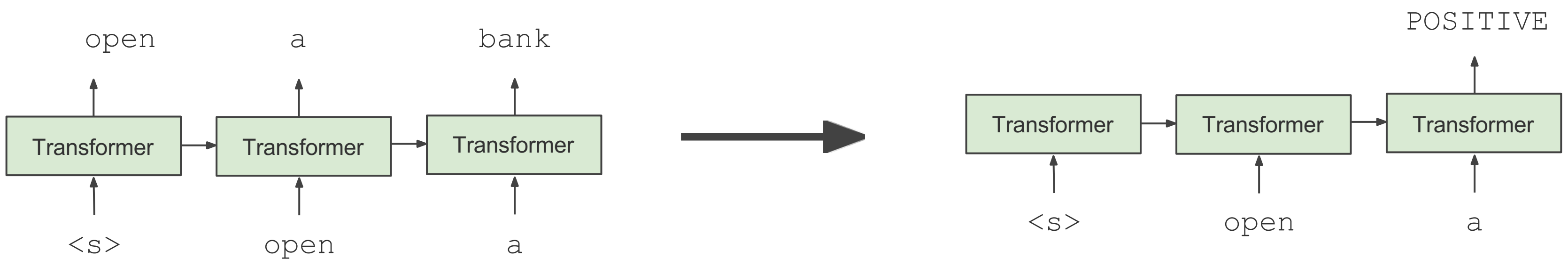
Courtesy: Devlin 2019

Contextual Representations

- *Improving Language Understanding by Generative Pre-Training*, OpenAI, 2018

Train Deep (12-layer) Transformer LM

Fine-tune on Classification Task



Courtesy: Devlin 2019

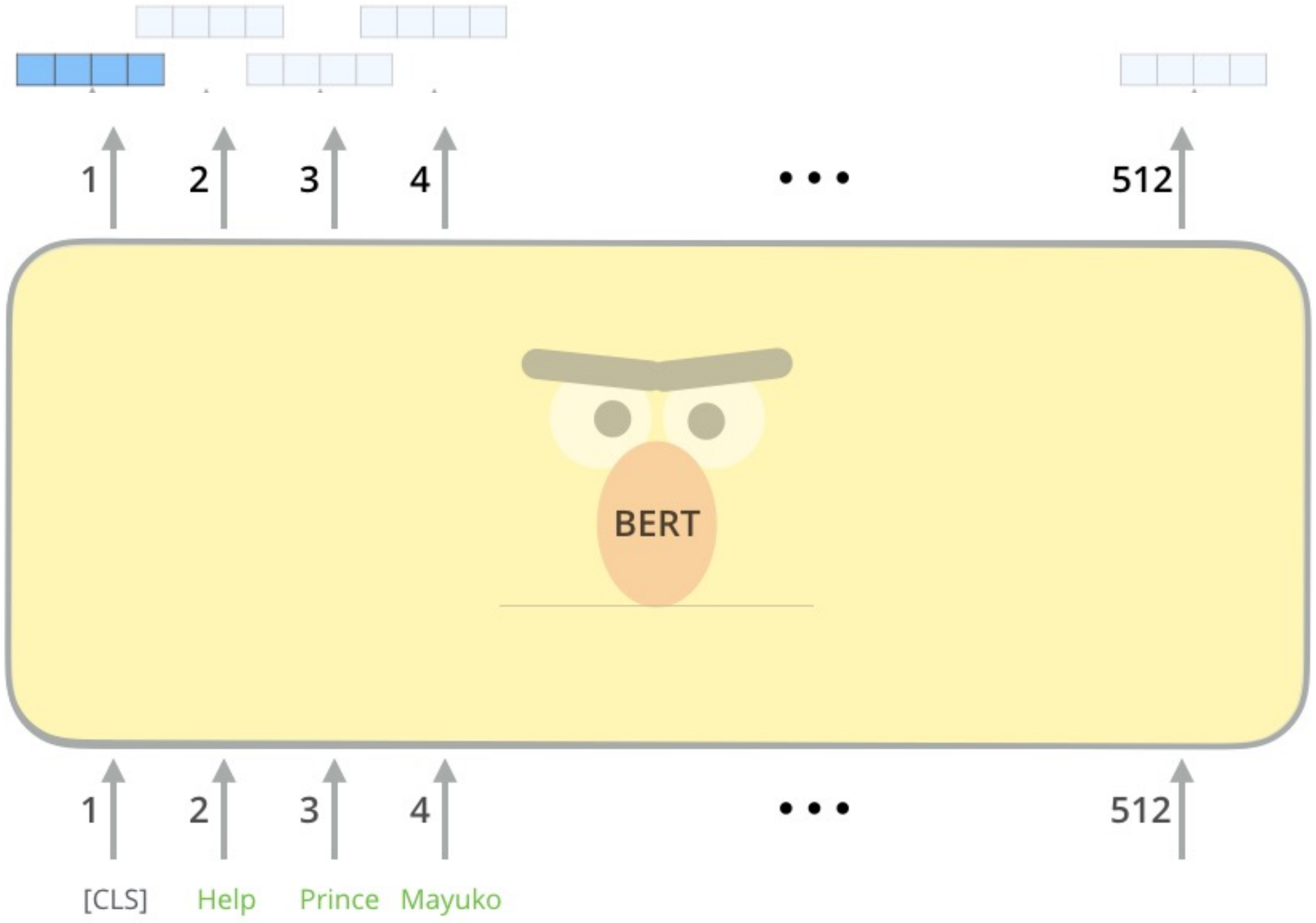
Problem with Previous Methods

- **Problem**: Language models only use left context *or* right context, but language understanding is bidirectional.

courtesy: Devlin 2019

BERT

- BERT: A bidirectional model to extract contextual word embedding



BERT: Pre-training Procedure

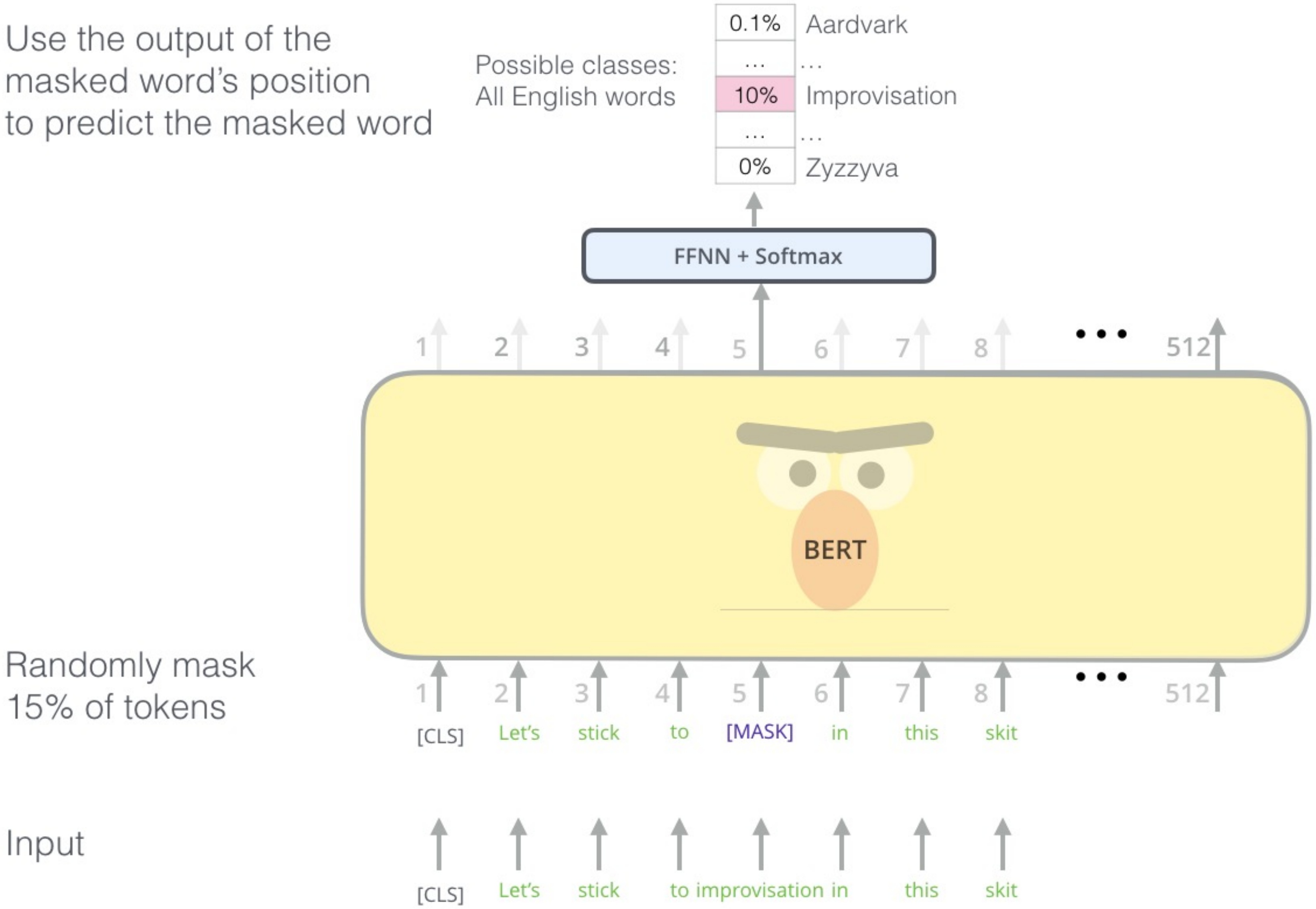
- Dataset:
  - Wikipedia (2.5B words) + a collection of free ebooks (800M words)

BERT: Pre-training Procedure

- Dataset:
  - Wikipedia (2.5B words) + a collection of free ebooks (800M words)
- Training procedure
  - **masked language model** (masked LM)
    - Masks some percent of words from the input and has to reconstruct those words from context

BERT: Pre-training Procedure

- Masked LM



BERT: Pre-training Procedure

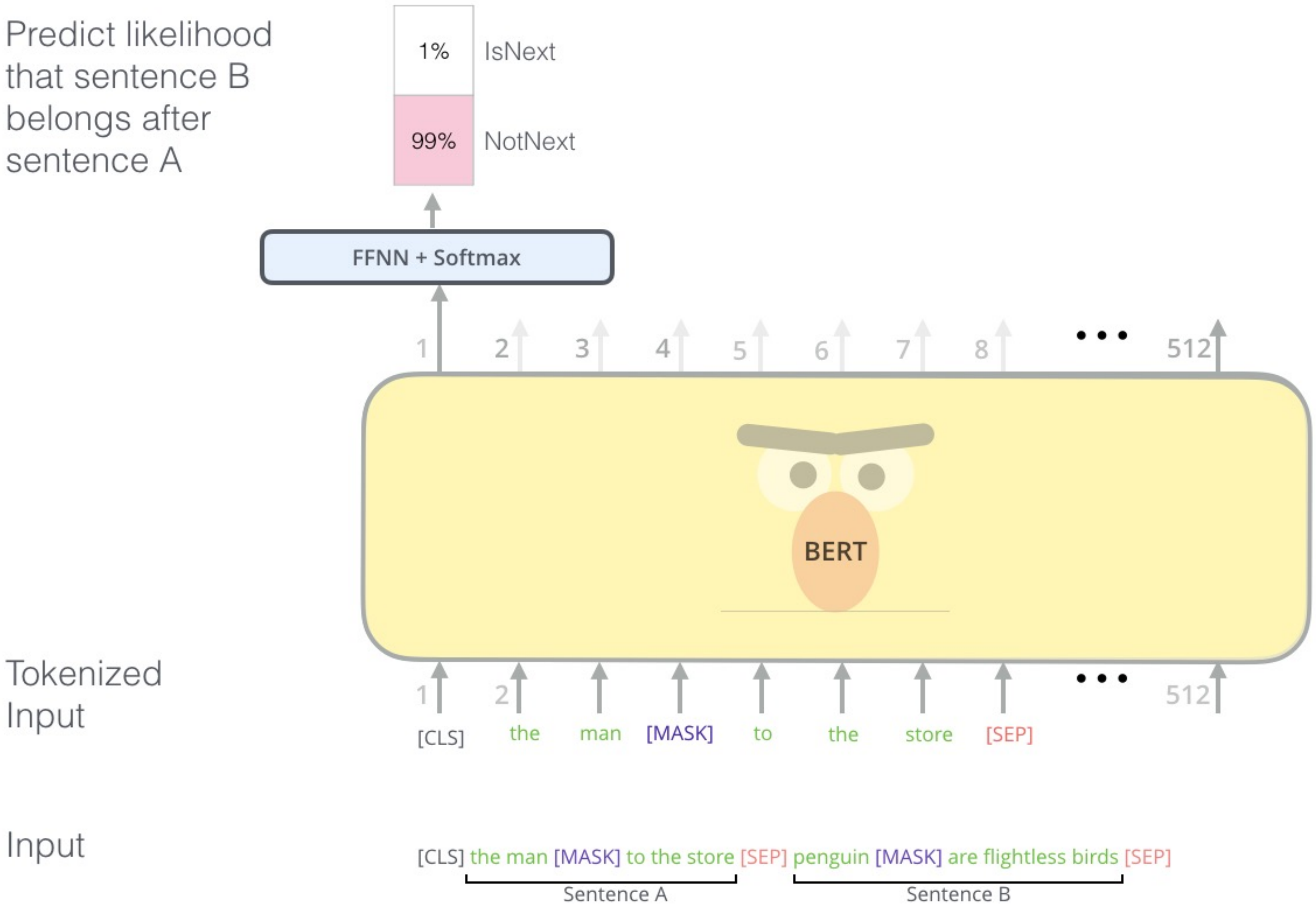
- Masked LM
- 15% masking:
  - Too little masking: Too expensive to train (few supervision signals per example)
  - Too much masking: Not enough context
- Problem: Mask token never seen at fine-tuning
- Solution: don't replace with `[MASK]` 100% of the time. Instead:
- 80% of the time, replace with `[MASK]`
  - `went to the store → went to the [MASK]`
- 10% of the time, replace random word
  - `went to the store → went to the running`
- 10% of the time, keep same
  - `went to the store → went to the store`

BERT: Pre-training Procedure

- Dataset:
  - Wikipedia (2.5B words) + a collection of free ebooks (800M words)
- Training procedure
  - **masked language model** (masked LM)
    - Masks some percent of words from the input and has to reconstruct those words from context
  - **Two-sentence task**
    - To understand relationships between sentences
    - Concatenate two sentences A and B and predict whether B actually comes after A in the original text

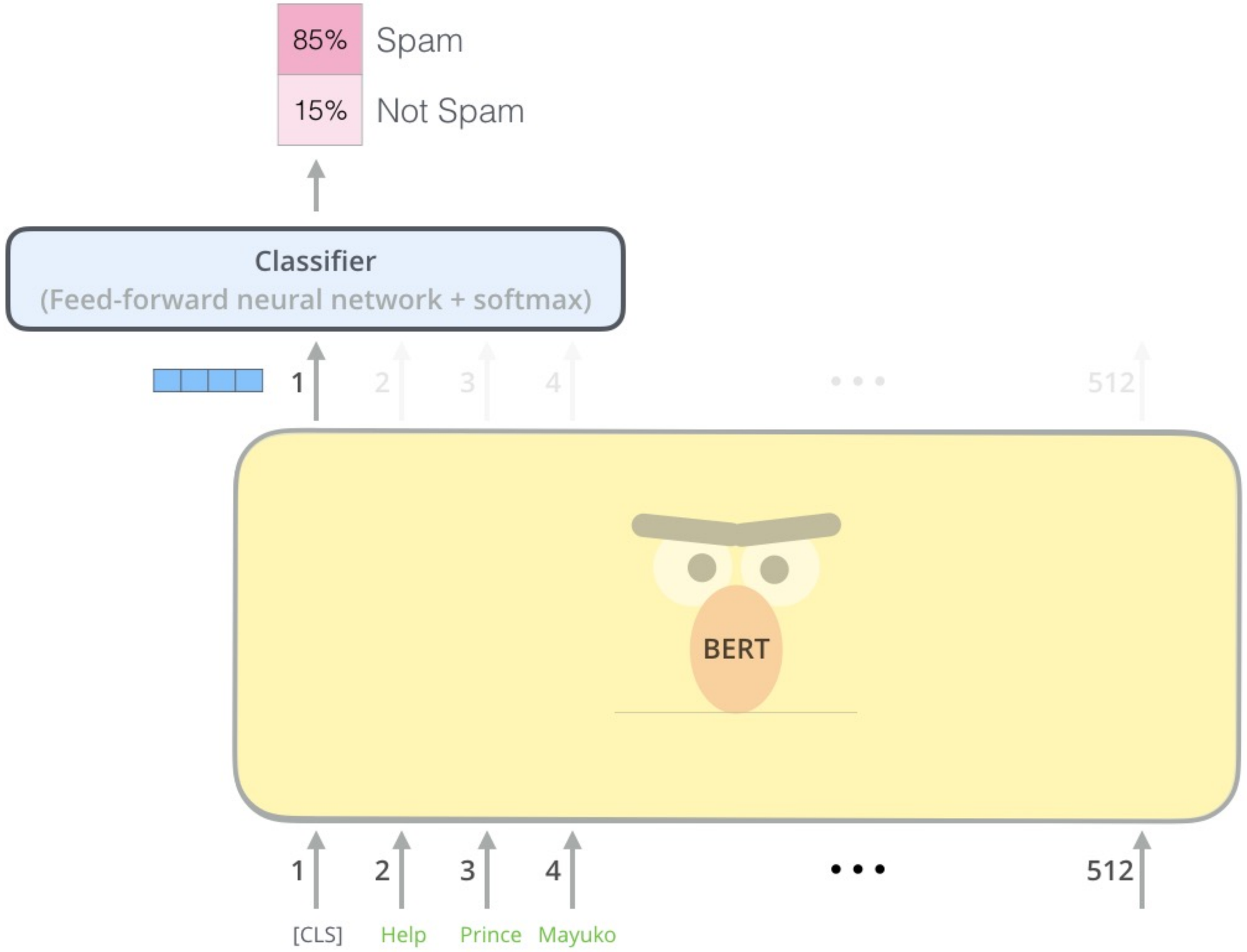
BERT: Pre-training Procedure

- Two sentence task

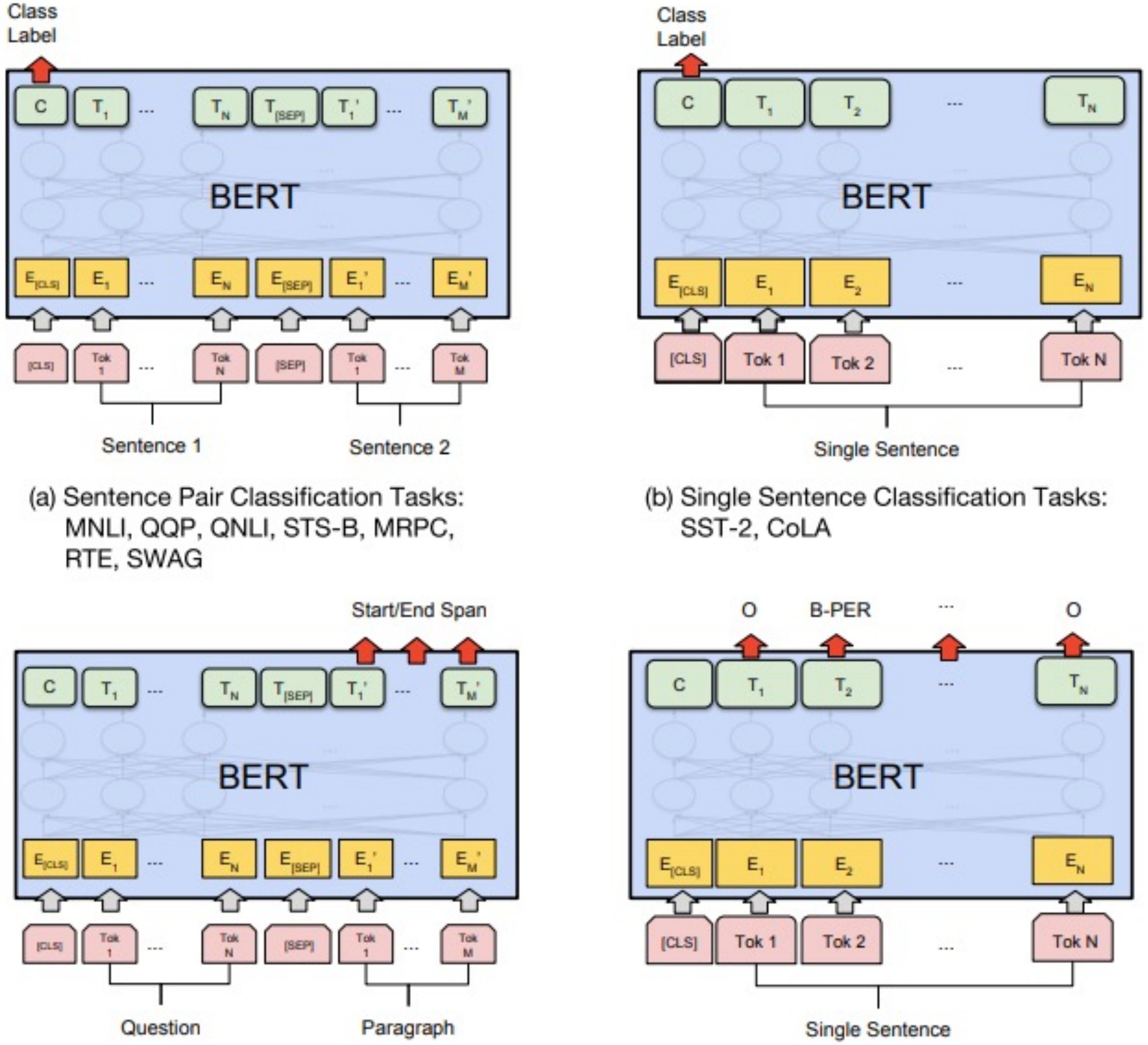


BERT: Downstream Fine-tuning

- Use BERT for sentence classification



BERT: Downstream Fine-tuning



(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

BERT Results

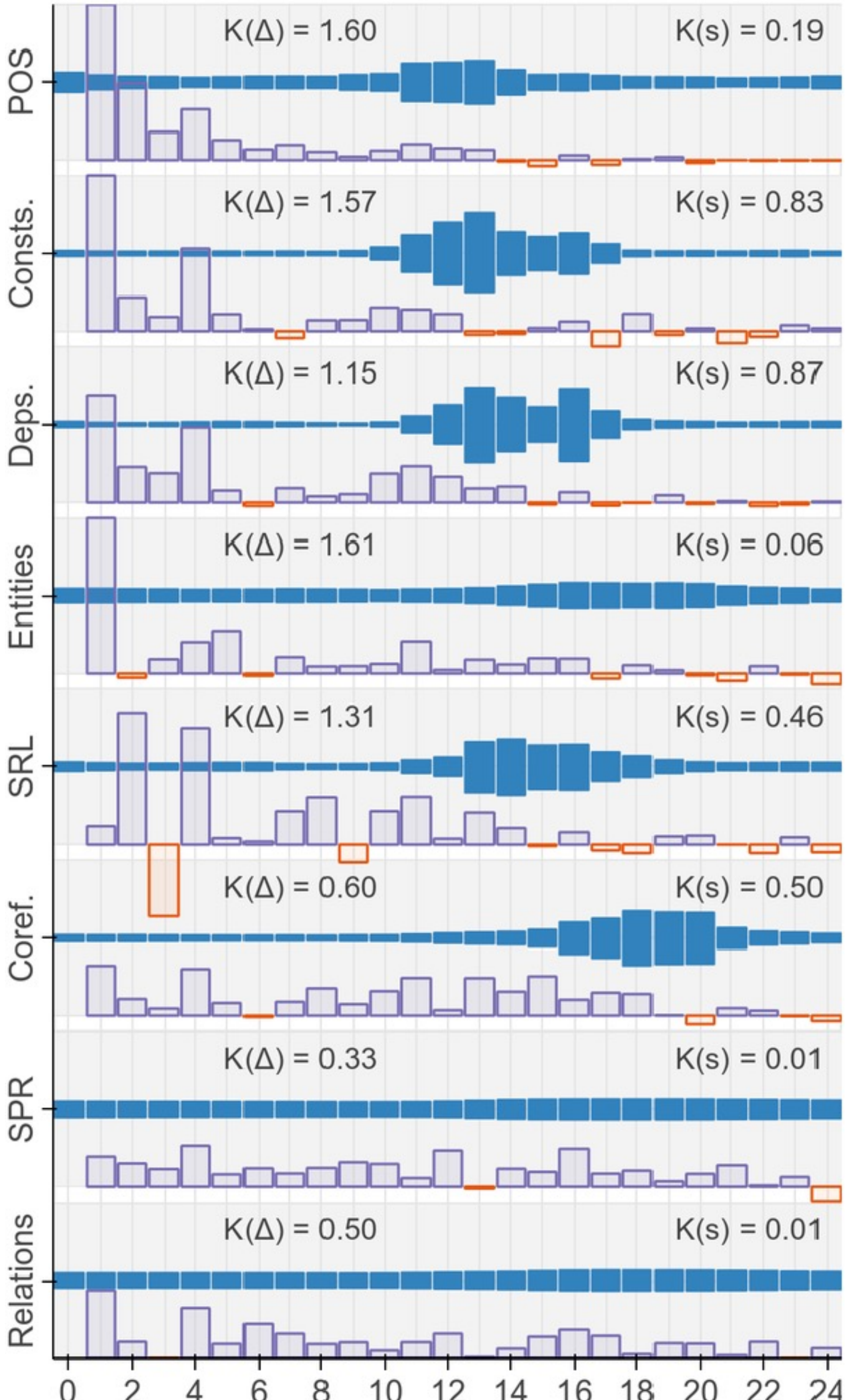
- Huge improvements over SOTA on 12 NLP task

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	79.9	90.4	36.0	73.3	84.9	56.8	71.0
OpenAI GPT	82.1/81.4	70.3	88.1	91.3	45.4	80.0	82.3	56.0	75.2
$\text{BERT}_{\text{BASE}}$	84.6/83.4	71.2	90.1	93.5	52.1	85.8	88.9	66.4	79.6
$\text{BERT}_{\text{LARGE}}$	**86.7/85.9**	**72.1**	**91.1**	**94.9**	**60.5**	**86.5**	**89.3**	**70.1**	**81.9**

Table 1: GLUE Test results, scored by the GLUE evaluation server. The number below each task denotes the number of training examples. The "Average" column is slightly different than the official GLUE score, since we exclude the problematic WNLI set. OpenAI GPT = (L=12, H=768, A=12); $\text{BERT}_{\text{BASE}}$ = (L=12, H=768, A=12); $\text{BERT}_{\text{LARGE}}$ = (L=24, H=1024, A=16). BERT and OpenAI GPT are single-model, single task. All results obtained from https://gluebenchmark.com/leaderboard and https://blog.openai.com/language-unsupervised/.

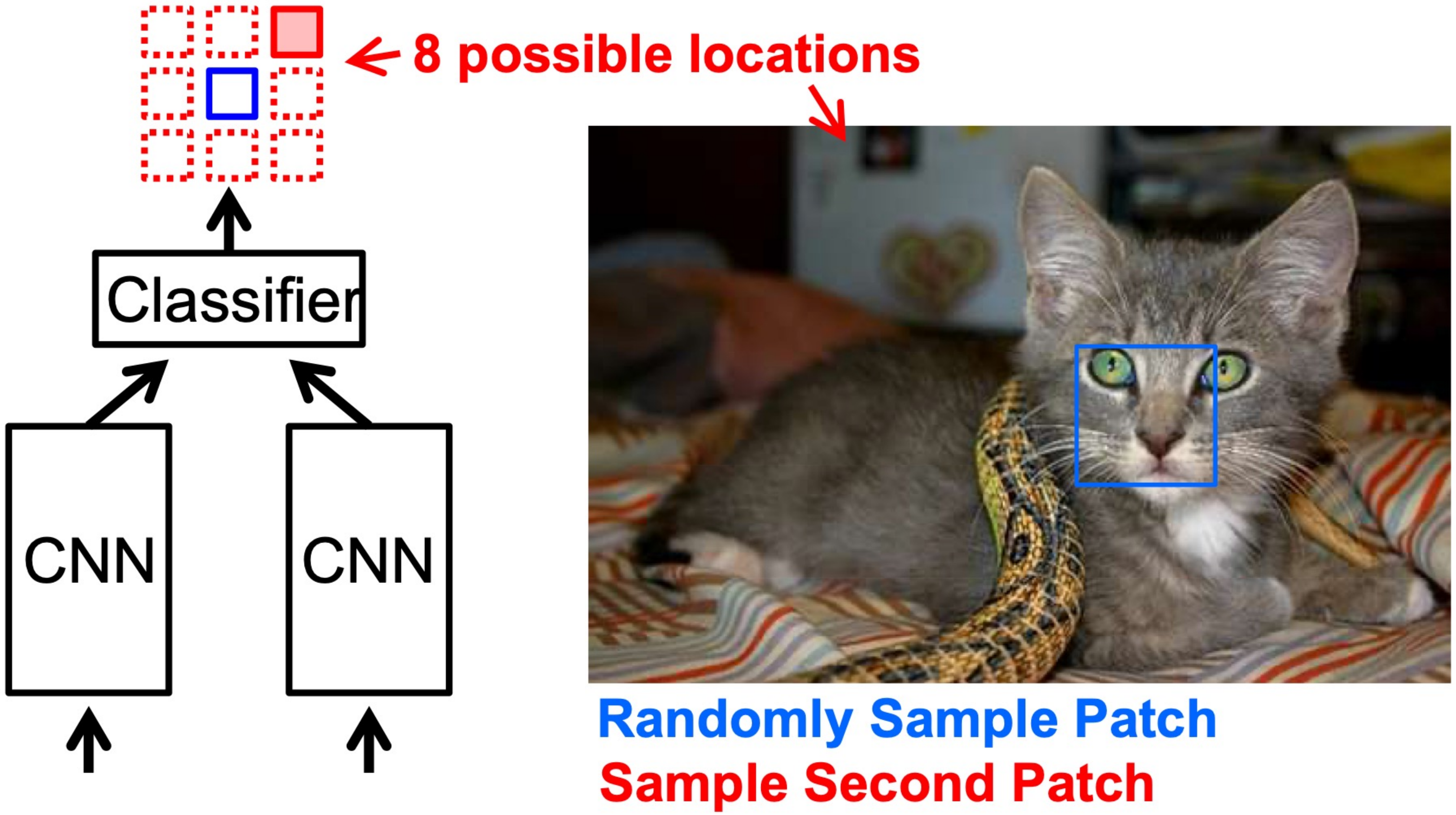
Analysis

- *BERT Rediscovers the Classical NLP Pipeline.* Tenney et al., 2019



SSL from Images, EX (I): relative positioning

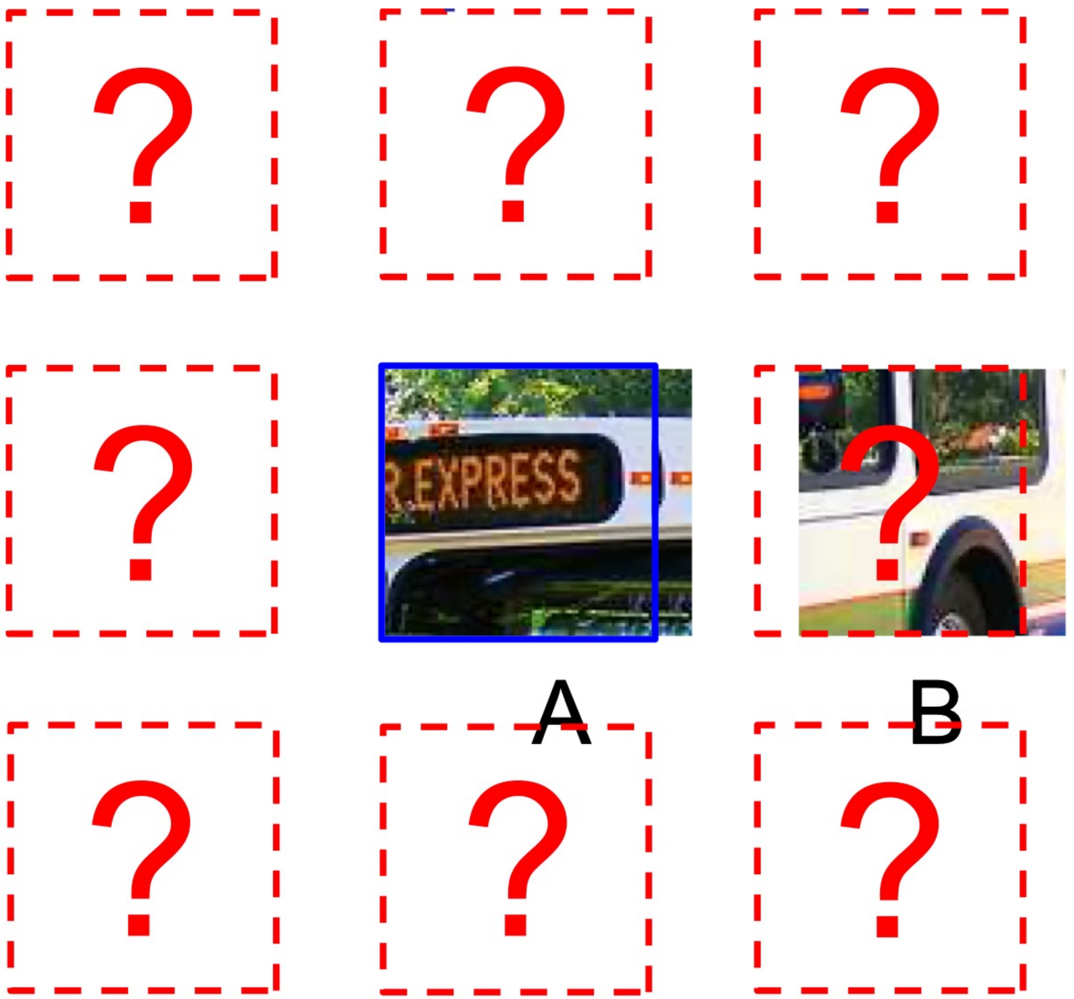
Train network to predict relative position of two regions in the same image



Unsupervised visual representation learning by context prediction,
Carl Doersch, Abhinav Gupta, Alexei A. Efros, ICCV 2015

[Courtesy: Zisserman “Self-supervised Learning”]

SSL from Images, EX (I): relative positioning



Unsupervised visual representation learning by context prediction,
Carl Doersch, Abhinav Gupta, Alexei A. Efros, ICCV 2015

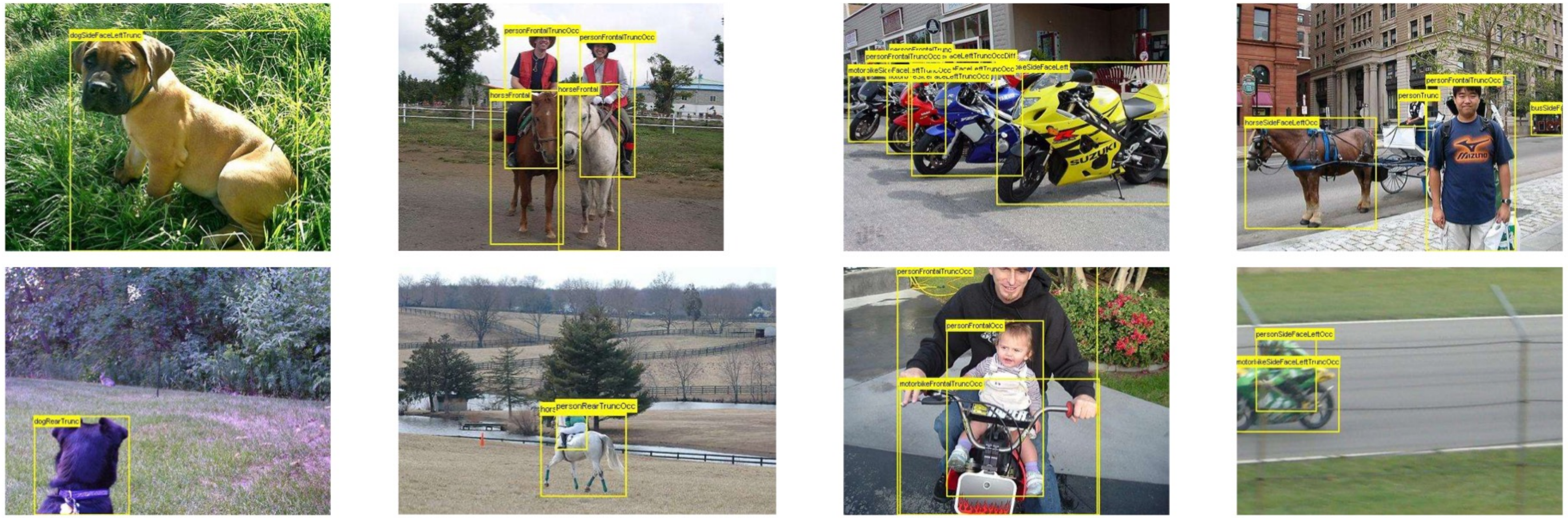
[Courtesy: Zisserman "Self-supervised Learning"]

SSL from Images, EX (I): relative positioning

Evaluation: PASCAL VOC Detection

- 20 object classes (car, bicycle, person, horse …)
- Predict the bounding boxes of all objects of a given class in an image (if any)

Dog | Horse | Motorbike | Person



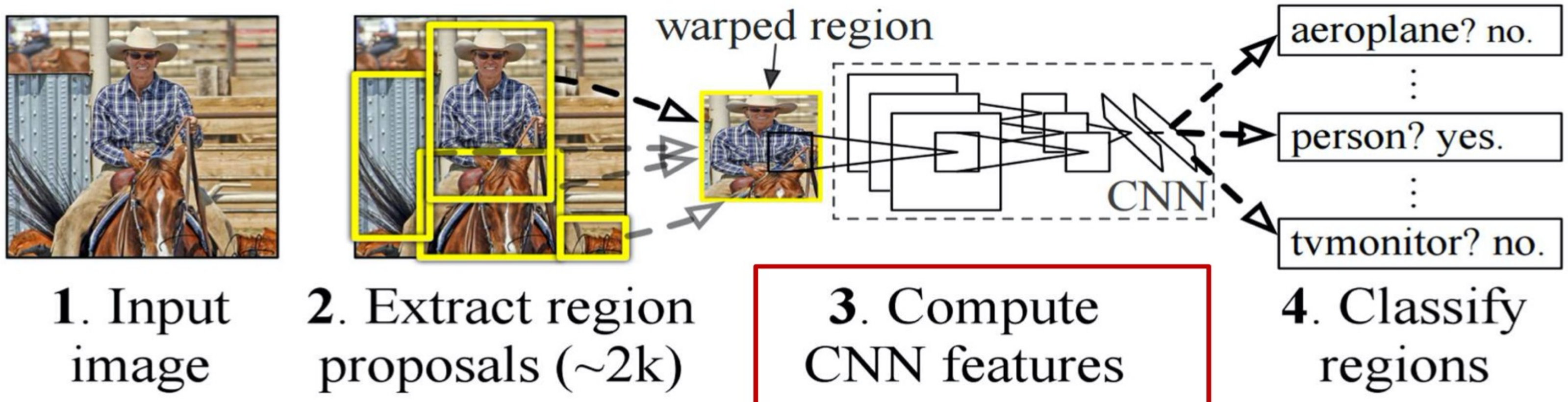
[Courtesy: Zisserman “Self-supervised Learning”]

SSL from Images, EX (I): relative positioning

Evaluation: PASCAL VOC Detection

- Pre-train CNN using self-supervision (no labels)
- Train CNN for detection in R-CNN object category detection pipeline

R-CNN



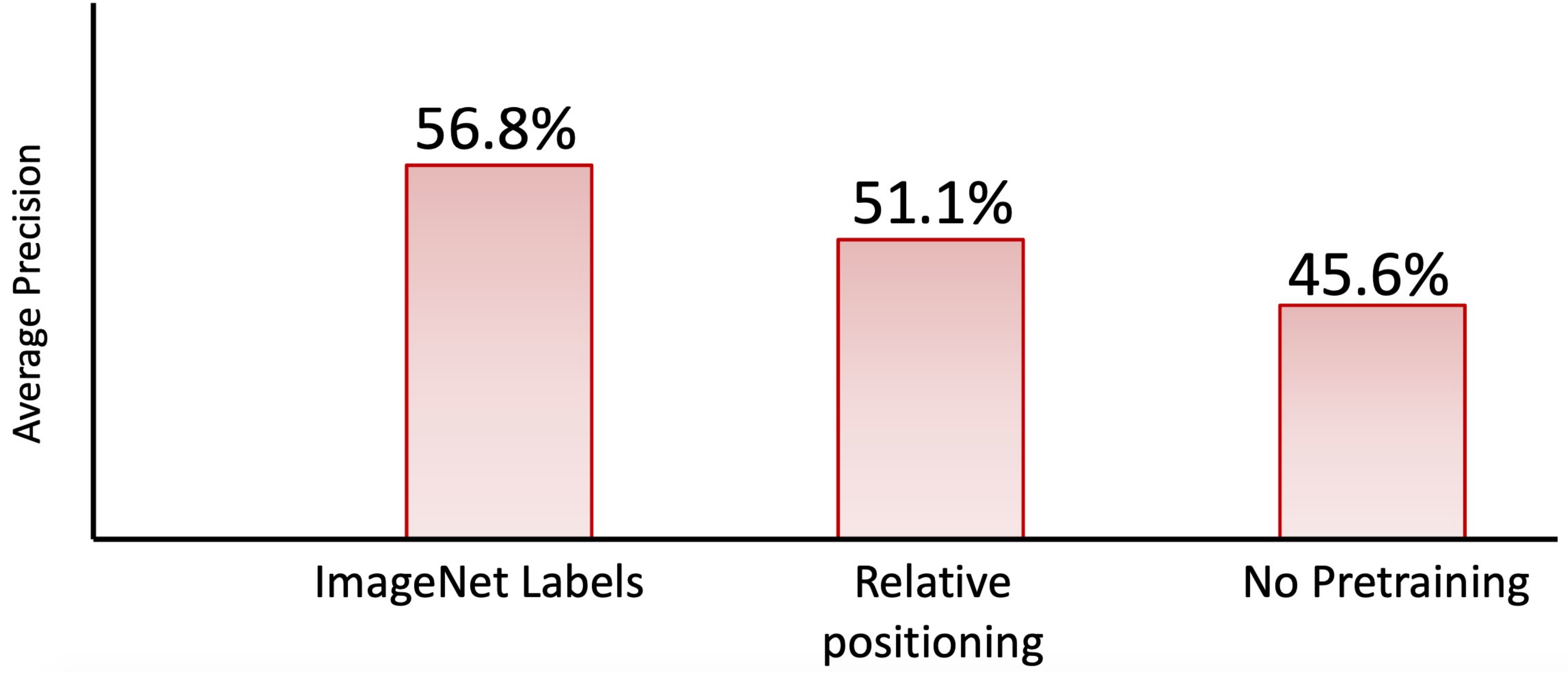
Pre-train on relative-position task, w/o labels

[Girshick et al. 2014]

[Courtesy: Zisserman "Self-supervised Learning"]

SSL from Images, EX (I): relative positioning

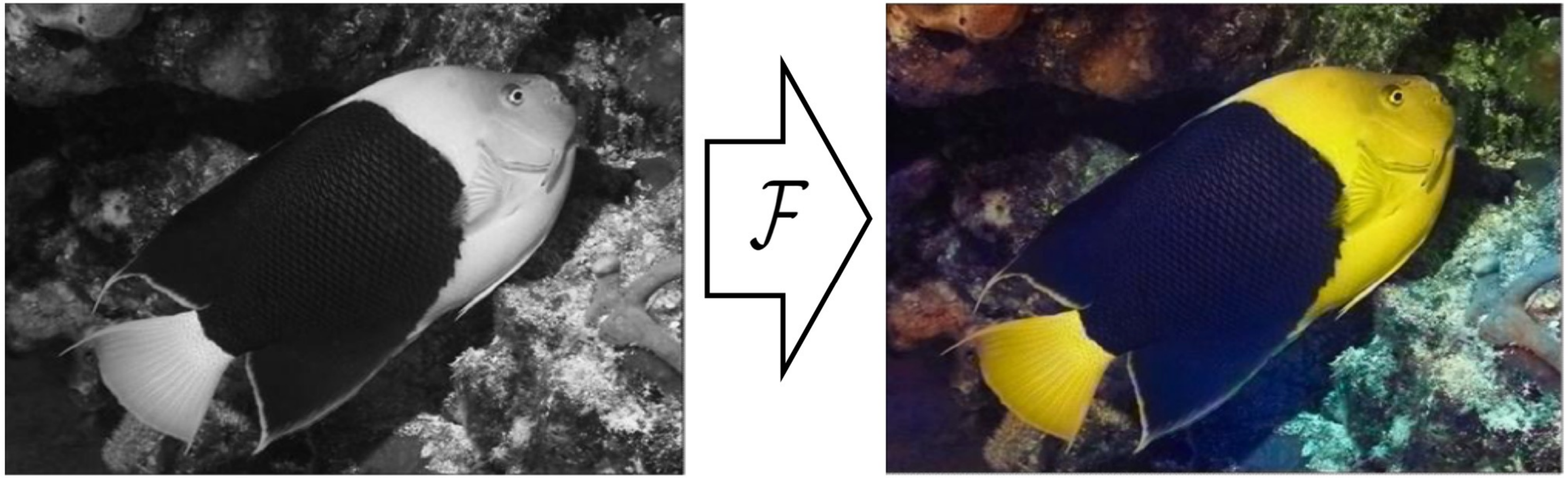
Evaluation: PASCAL VOC Detection



[Courtesy: Zisserman “Self-supervised Learning”]

SSL from Images, EX (II): colorization

Train network to predict pixel colour from a monochrome input

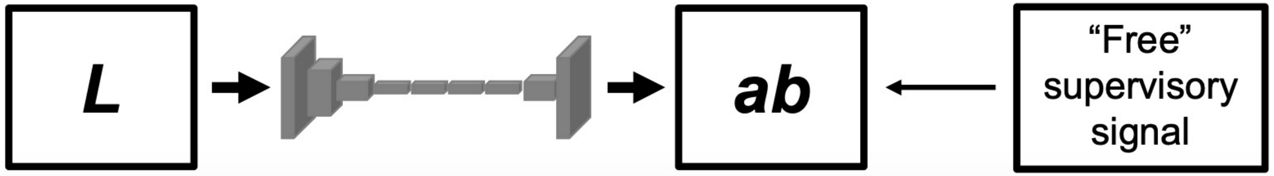


Grayscale image: *L* channel

$$\mathbf{X} \in \mathbb{R}^{H \times W \times 1}$$

Concatenate (L,ab)

$$(\mathbf{X}, \hat{\mathbf{Y}})$$



[Courtesy: Zisserman "Self-supervised Learning"]

Colorful Image Colorization, Zhang et al., ECCV 2016

SSL from Images, EX (II): colorization

Train network to predict pixel colour from a monochrome input

[Courtesy: Zisserman "Self-supervised Learning"]

Colorful Image Colorization, Zhang et al., ECCV 2016

SSL from Images, EX (III): exemplar networks

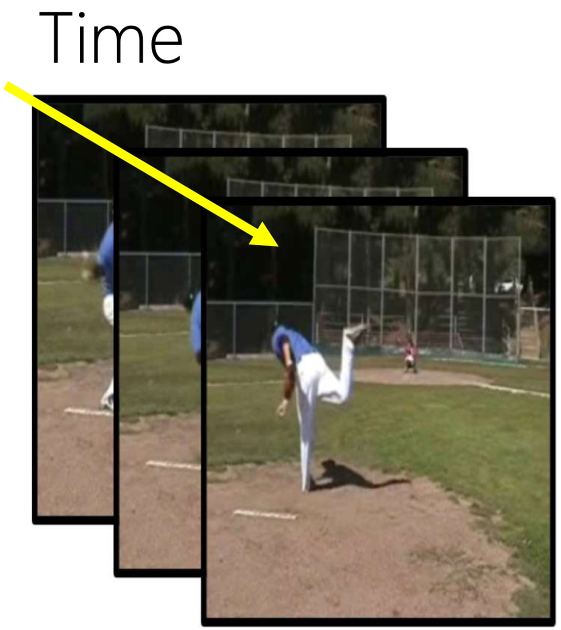
- Exemplar Networks (Dosovitskiy et al., 2014)
- Perturb/distort image patches, e.g. by cropping and affine transformations
- Train to classify these exemplars as same class

[Courtesy: Zisserman "Self-supervised Learning"]

SSL from Videos

Three example tasks:

- Video sequence order
  - Sequential Verification: Is this a valid sequence?



“Sequence” of data

[Courtesy: Zisserman “Self-supervised Learning”]

Wei et al., 2018 Arrow of Time

SSL from Videos

Three example tasks:

- Video sequence order
  - Sequential Verification: Is this a valid sequence?
- Video direction
  - Predict if video playing forwards or backwards

[Courtesy: Zisserman “Self-supervised Learning”]

Wei et al., 2018 Arrow of Time

SSL from Videos

Three example tasks:

- Video sequence order
  - Sequential Verification: Is this a valid sequence?
- Video direction
  - Predict if video playing forwards or backwards
- Video tracking
  - Given a color video, colorize all frames of a gray scale version using a reference frame

[Courtesy: Zisserman “Self-supervised Learning”]

Vondrick et al., 2018

Key Takeaways

- Self supervision learning
  - Predicting any part of the observations given any available information
  - The prediction task forces models to learn semantic representations
  - Massive/unlimited data supervisions
- SSL for text:
  - Language models: next word prediction
  - BERT text representations: masked language model (MLM)
- SSL for images/videos:
  - Various ways of defining the prediction task

Questions?