Lecture Notes for

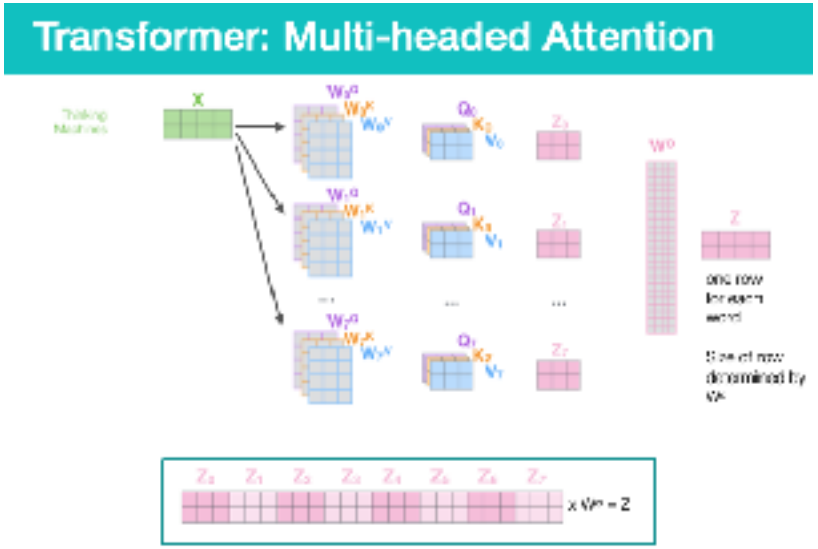
Neural Networks and Machine Learning

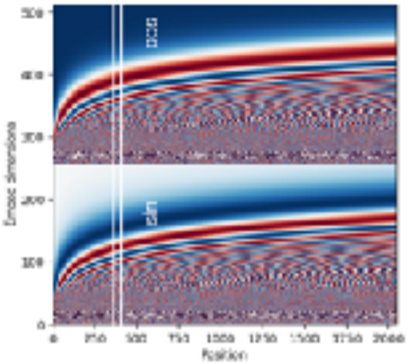
Vision Transformers
Self-supervised Learning
Consistency Loss

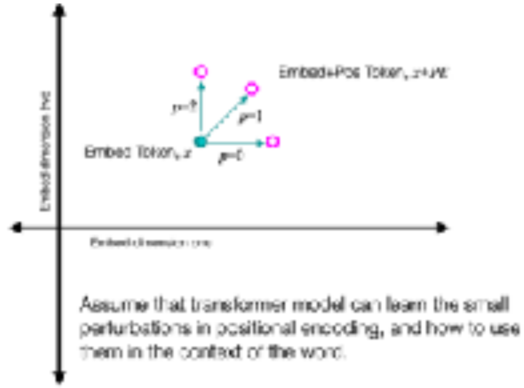
Logistics and Agenda

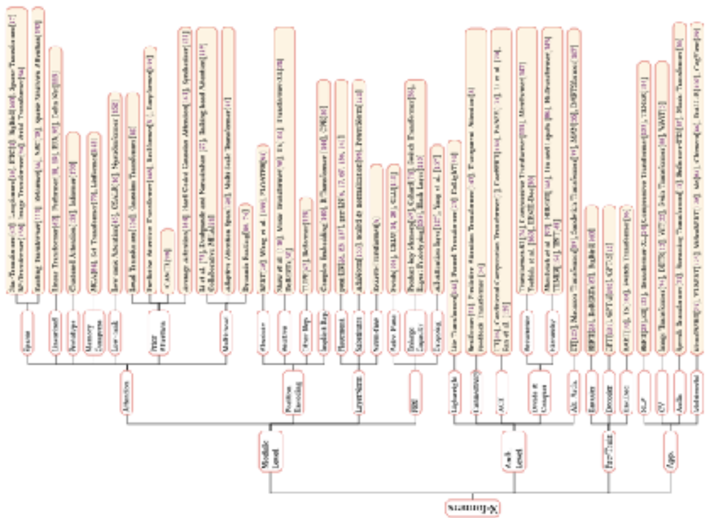
- Logistics
  - None!
- Agenda
  - Vision Transformers
  - Self Supervised Learning and Consistency Loss
- Next Time:
  - Paper Presentation: *Language Models are Few Shot Learners*
  - Multi-modal Learning
    - Techniques
    - Applications and domains
  - Multi-Task and Demo

Last Time: Transformers







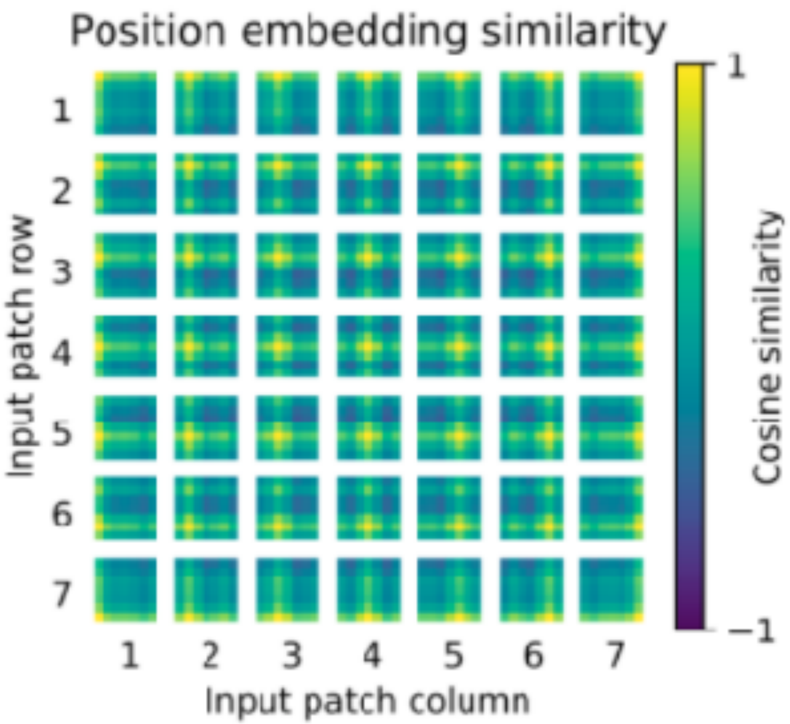


51

Vision Transformers

Vision Transformers

- Divide image into patches
  - Treat each patch as something to encode separately
  - Flatten each patch
  - Put through dense layer
- Add positional encoding based on position of patch
  - for 7x7 patch, there are 49 positions
- Put into transformer. Same as text transformers …
- **But you need a lot of data**
  - 14M or more images seems to be sweet spot



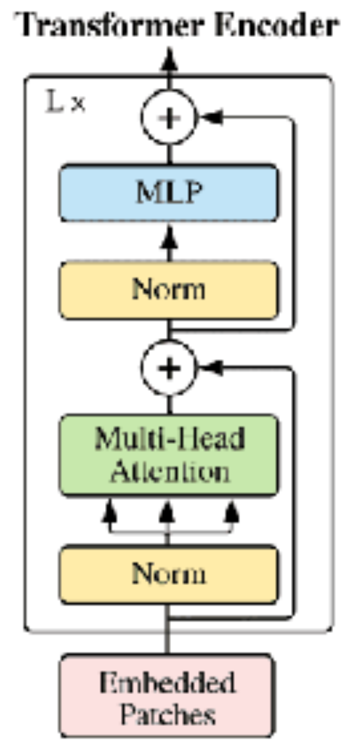
Vision Transformers Video

https://ai.googleblog.com/2020/12/transformers-for-image-recognition-at.html?m=1

ViT Architectures

- *D* is size of patch embedding
- Uses skip connections (all size *D*)
- Multi-headed self attention (MSA) takes *D* input `patch_embed + pos_embed`
- Main difference in architectures
  - *L* blocks used (*i.e.*, “layers”)
  - *H* heads in each layer (*i.e.*, “heads”)
  - MLP head is final classifier

$$
\begin{aligned}
\mathbf{z}_0 &= [\mathbf{x}_{\text{class}}; \mathbf{x}_p^1\mathbf{E}; \mathbf{x}_p^2\mathbf{E}; \cdots; \mathbf{x}_p^N\mathbf{E}] + \mathbf{E}_{pos}, \\
\mathbf{z}'_\ell &= \mathrm{MSA}(\mathrm{LN}(\mathbf{z}_{\ell-1})) + \mathbf{z}_{\ell-1}, \\
\mathbf{z}_\ell &= \mathrm{MLP}(\mathrm{LN}(\mathbf{z}'_\ell)) + \mathbf{z}'_\ell, \\
\mathbf{y} &= \mathrm{LN}(\mathbf{z}_L^0)
\end{aligned}
$$



Model	Layers	Hidden size D	MLP size	Heads	Params
ViT-Base	12	768	3072	12	86M
ViT-Large	24	1024	4096	16	307M
ViT-Huge	32	1280	5120	16	632M

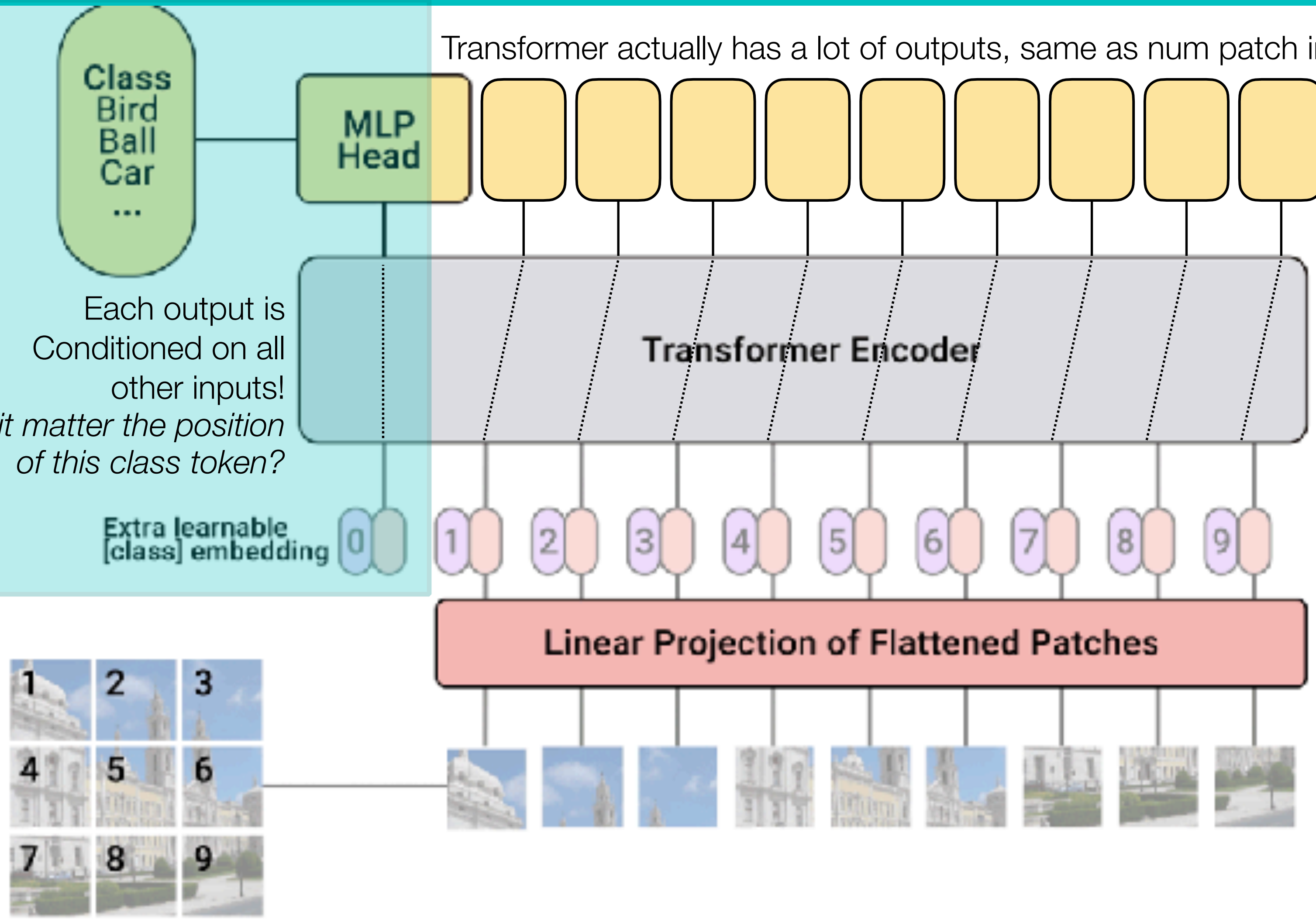
ResNet50: 23M

Dosovitskiy *et al.* (2021) An image is worth 16x16 words: Transformers for Image Recognition at Scale https://arxiv.org/abs/2010.11929

What is the learnable class embedding?

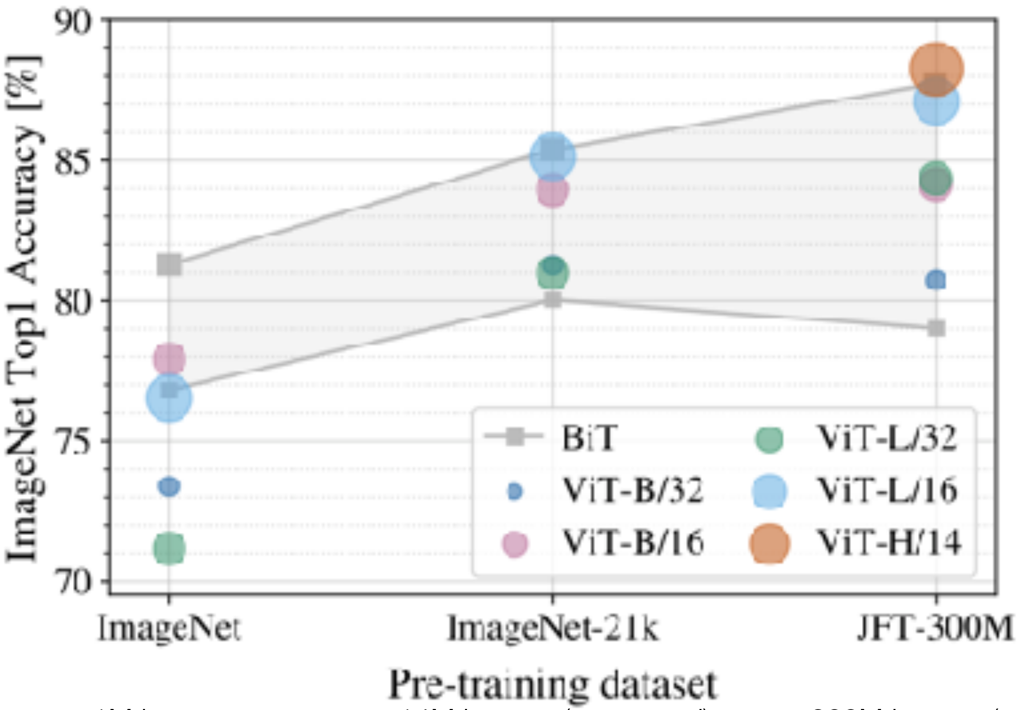
Transformer actually has a lot of outputs, same as num patch inputs

Each output is Conditioned on all other inputs!
Does it matter the position of this class token?



Do they work?

- Yes, but good luck getting weights for them or training them, even with the SuperPOD
- Less than 14M images for pre-training? Use ResNet.



~1M Images
1k classes

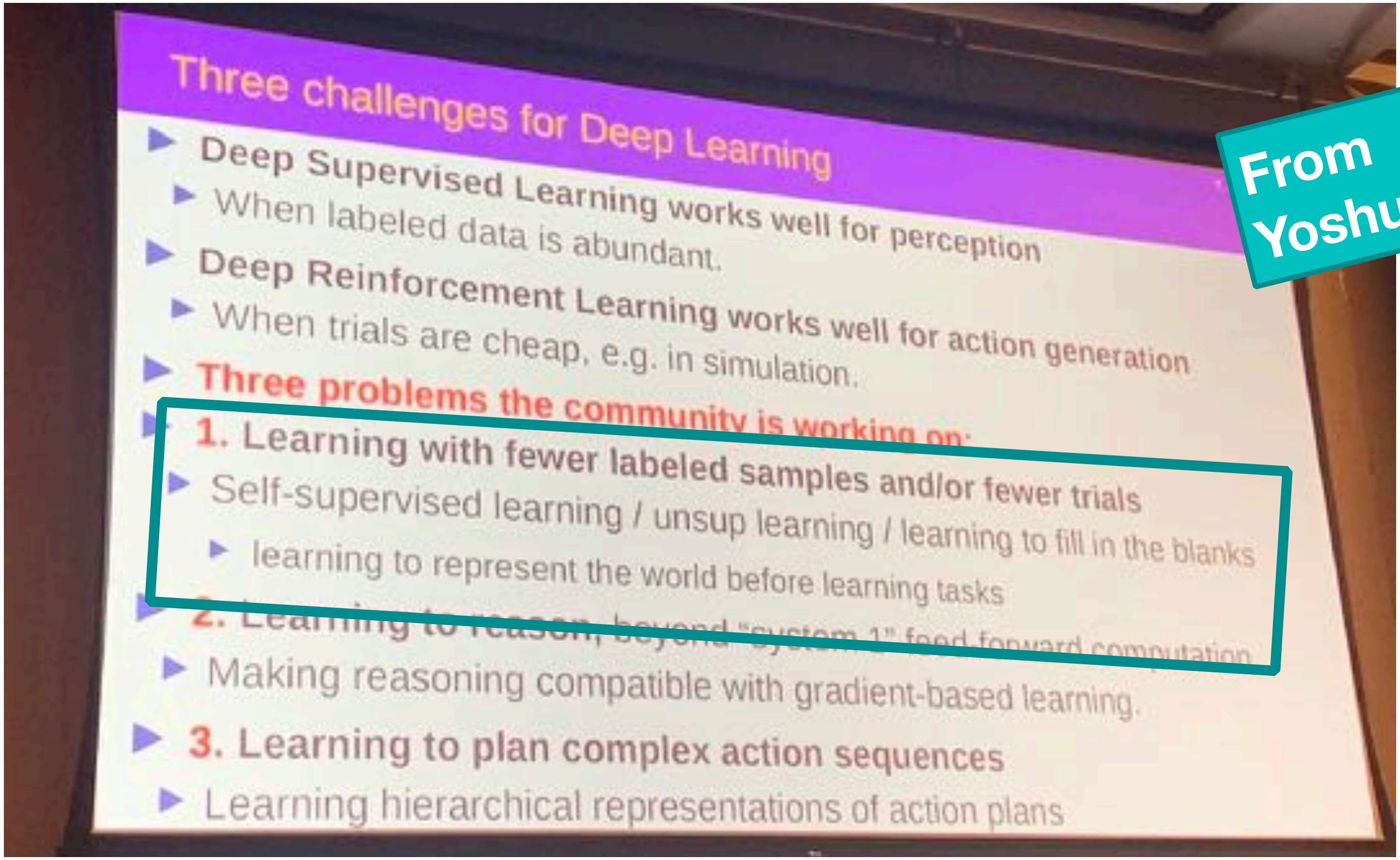
~14M Images (part. crawl)
21k classes

~300M Images (crawled)
18k classes

Transfer Learning From Huge ViT

	ViT-H	Previous SOTA
ImageNet	88.55	88.5
ImageNet-ReaL	90.72	90.55
Cifar-10	99.50	99.37
Cifar-100	94.55	93.51
Pets	97.56	96.62
Flowers	99.68	99.63

Self-Supervised Learning

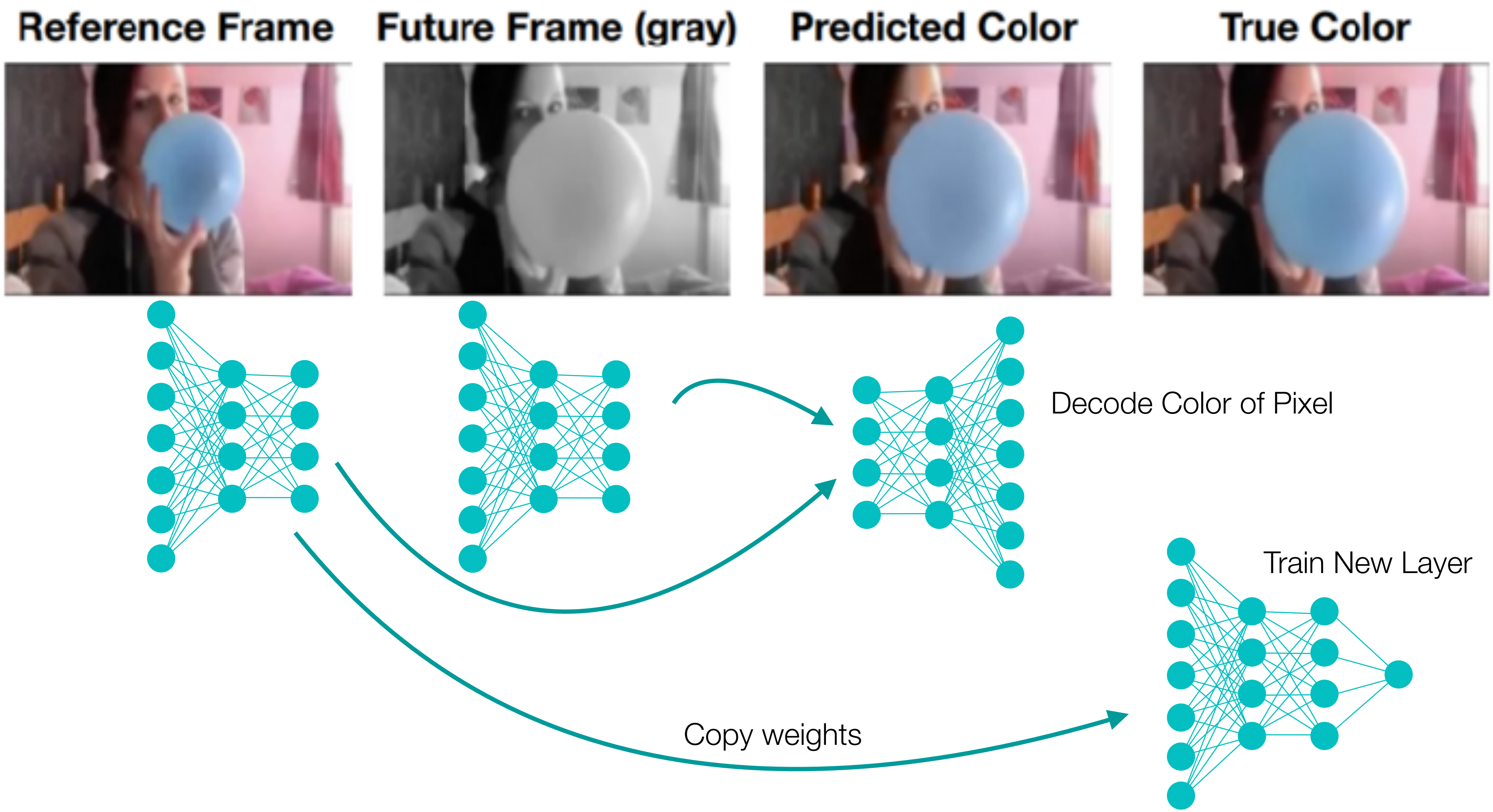


From Yoshua Bengio

Self-supervised Learning

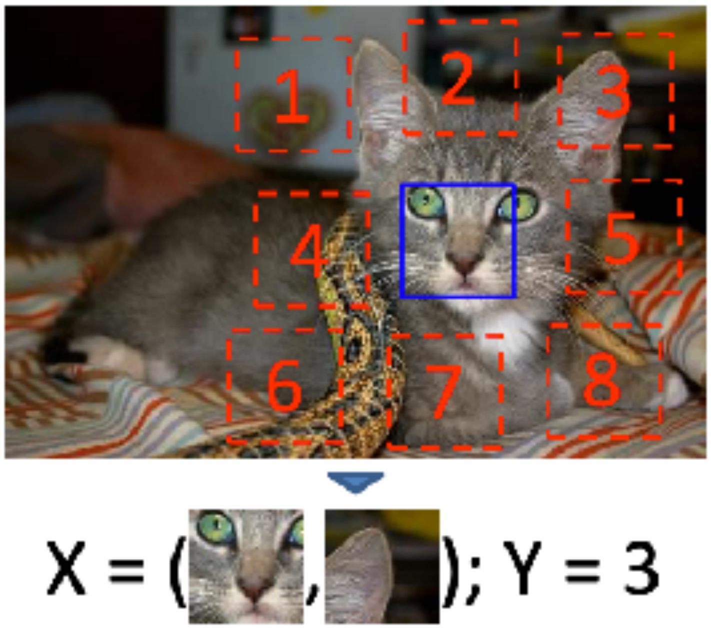
- **Problem**: deep learning is not sample efficient
- **Idea**: learn about the world before learning the task
- **New Problem**: how do we learn about the world?
- **Solution**: transfer learning on toy problem
  - 1. train on auxiliary task that is easy to label
  - 2. throw away anything specific to auxiliary task
  - 3. train new network with task of interest, transferring knowledge (downstream task)
  - 4. profit

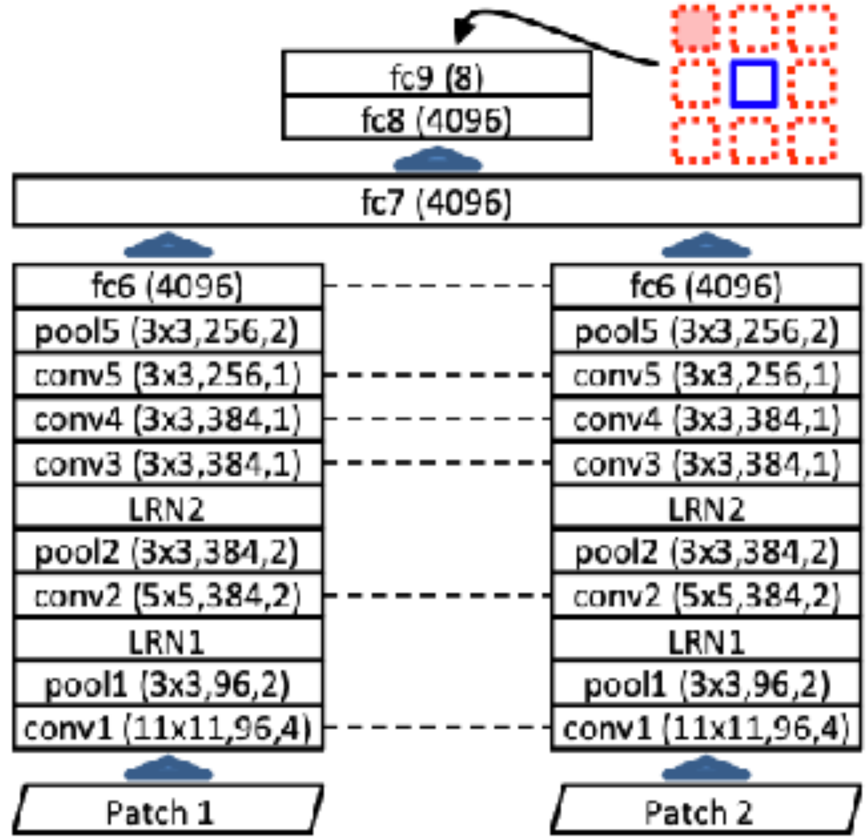
Examples of Self Supervised Learning



https://www.fast.ai/2020/01/13/self_supervised/

Examples of Self Supervised Learning





Unsupervised Visual Representation Learning by Context Prediction

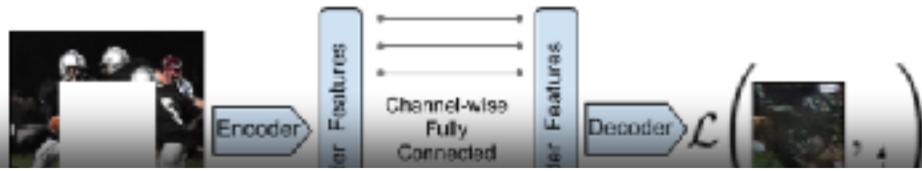
Carl Doersch[1,2] Abhinav Gupta[1] Alexei A. Efros[2]

[1] School of Computer Science, Carnegie Mellon University

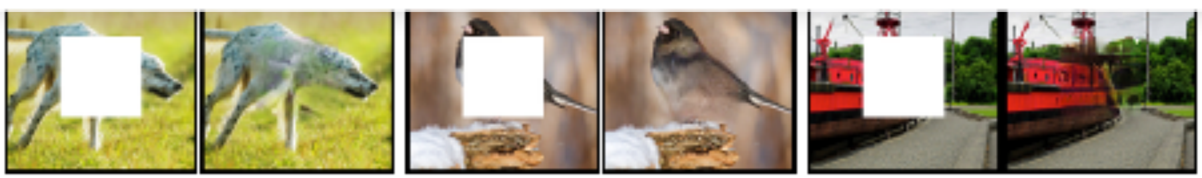
[2] Dept. of Electrical Engineering and Computer Science, University of California, Berkeley

https://www.fast.ai/2020/01/13/self_supervised/

Examples of Self Supervised Learning



Pretraining Method	Supervision	Pretraining time	Classification	Detection	Segmentation
ImageNet [26]	1000 class labels	3 days	78.2%	56.8%	48.0%
Random Gaussian	initialization	< 1 minute	53.3%	43.4%	19.8%
Autoencoder	-	14 hours	53.8%	41.9%	25.2%
Agrawal *et al.* [1]	egomotion	10 hours	52.9%	41.8%	-
Wang *et al.* [39]	motion	1 week	58.7%	47.4%	-
Doersch *et al.* [7]	relative context	4 weeks	55.3%	46.6%	-
Ours	context	14 hours	56.5%	44.5%	30.0%



Doesn't always work to increase performance…

Context Encoders: Feature Learning by Inpainting

Deepak Pathak Philipp Krähenbühl Jeff Donahue Trevor Darrell Alexei A. Efros
University of California, Berkeley

https://www.fast.ai/2020/01/13/self_supervised/

Consistency Loss

I'm from Canada, but live in the States now.

It took me a while to get used to writing boolean variables with an "Is" prefix, instead of the "Eh" suffix that Canadians use when programming.

For example:

```
MyObj.IsVisible

MyObj.VisibleEh
```

Unsupervised Consistency Loss

$$\min_{\mathbf{w}} \overbrace{\mathbf{E}_{\mathbf{x},y\in L}[-\log p_{\mathbf{w}}(y\,|\,\mathbf{x})]}^{\text{cross entropy}} + \lambda \overbrace{\mathscr{D}_{KL}\left(p_{\mathbf{w}}(y\,|\,\mathbf{x})\,||\,p_{\mathbf{w}}(y\,|\,\hat{\mathbf{x}})\right)}^{\text{consistency in augmentation}}$$

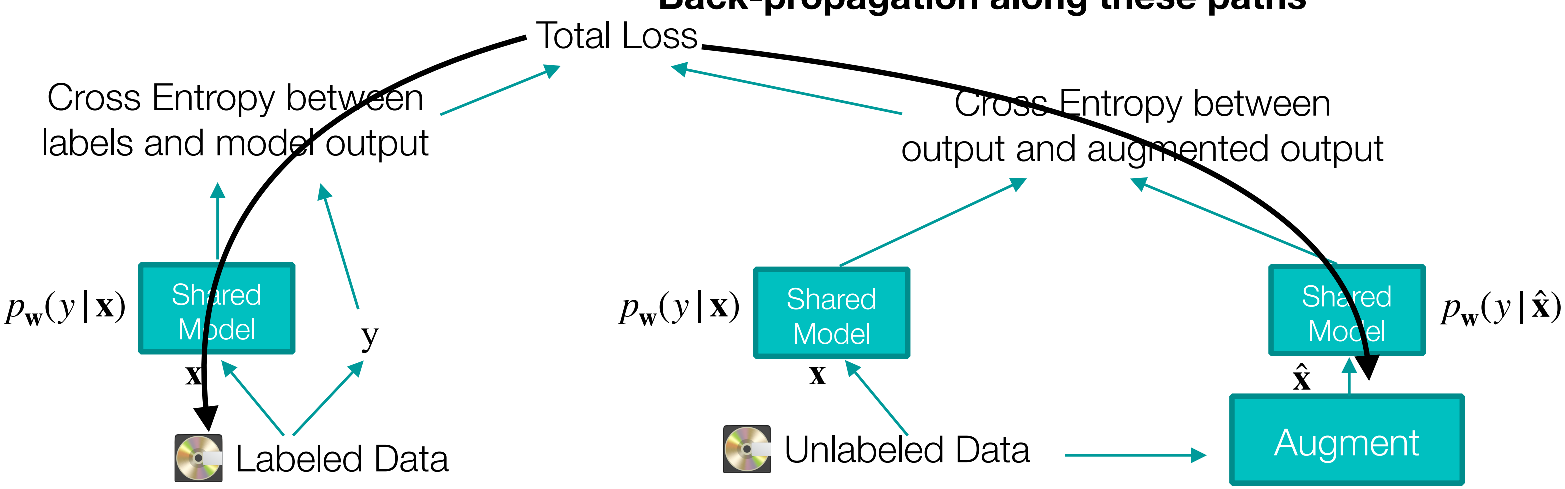
no back prop **yes** back prop

Neural Network approximates $p(y|\mathbf{x})$ by $\mathbf{w}$
Use labeled data to minimize network

Get accustomed to this notation

Sample new $\mathbf{x}$ from unlabeled pool with function q
function q is augmentation procedure
Minimize cross entropy of two models

Update Model with
Back-propagation along these paths



Unsupervised Data Augmentation (UDA) for Consistency Training, Xie et al., NeurIps 2019

Unsupervised Consistency Loss

$$\min_{\mathbf{w}} \overbrace{\mathbf{E}_{\mathbf{x},y\in L}[-\log p_{\mathbf{w}}(y\,|\,\mathbf{x})]}^{\text{cross entropy}} + \lambda \overbrace{\mathcal{D}_{KL}\left(p_{\mathbf{w}}(y\,|\,\mathbf{x})\,||\,p_{\mathbf{w}}(y\,|\,\hat{\mathbf{x}})\right)}^{\text{consistency in augmentation}}$$

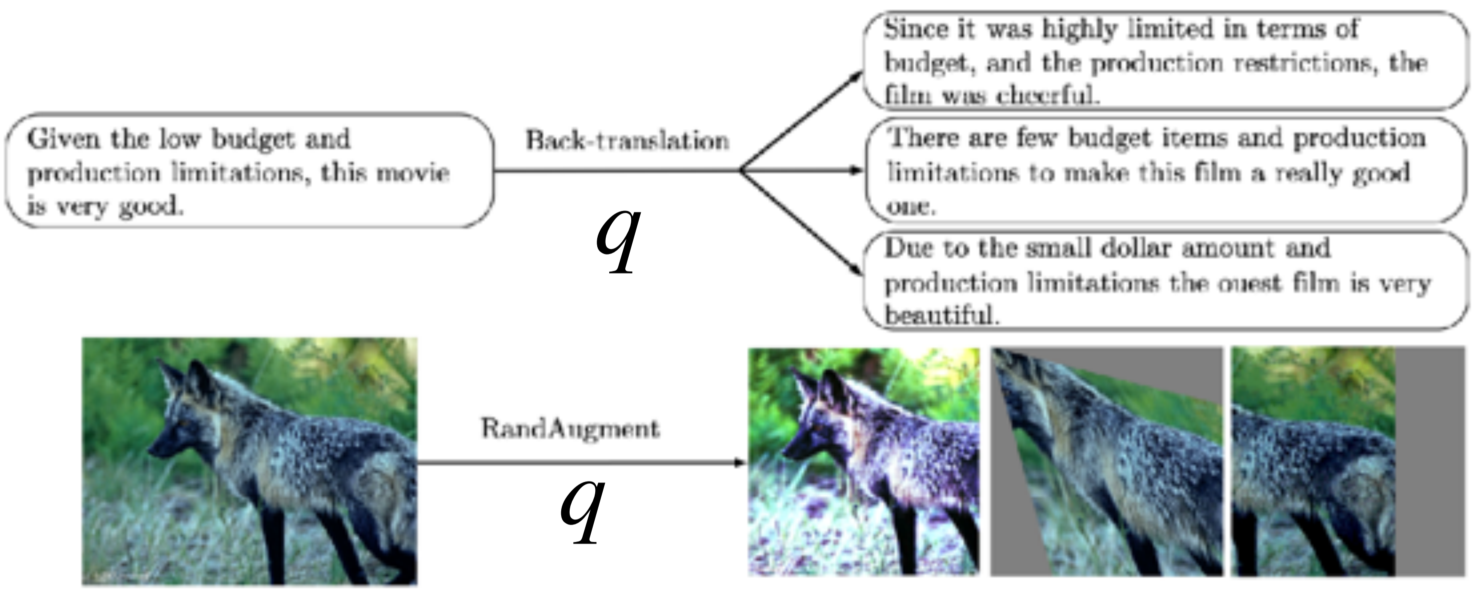


Figure 2: Augmented examples using back-translation and RandAugment.

Unsupervised Data Augmentation (UDA) for Consistency Training, Xie et al., NeurIps 2019

Unsupervised Consistency Loss (review)

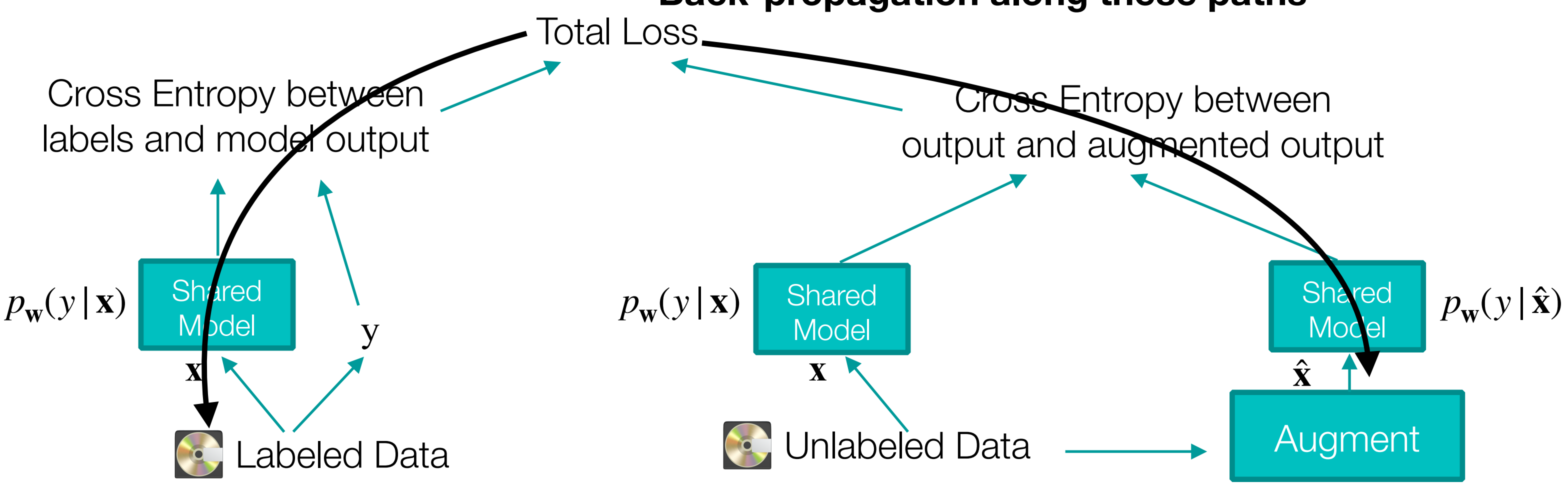
$$\min_{\mathbf{w}} \overbrace{\mathbf{E}_{\mathbf{x},y\in L}[-\log p_{\mathbf{w}}(y\,|\,\mathbf{x})]}^{\text{cross entropy}} + \lambda \overbrace{\mathscr{D}_{KL}\left(\underbrace{p_{\mathbf{w}}(y\,|\,\mathbf{x})}_{\textbf{no}\text{ back prop}}\,||\,\underbrace{p_{\mathbf{w}}(y\,|\,\hat{\mathbf{x}})}_{\textbf{yes}\text{ back prop}}\right)}^{\text{consistency in augmentation}}$$

Neural Network approximates $p(y|\mathbf{x})$ by $\mathbf{w}$
Use labeled data to minimize network

Get accustomed to this notation

Sample new $\mathbf{x}$ from unlabeled pool with function q
function q is augmentation procedure
Minimize cross entropy of two models

Update Model with
Back-propagation along these paths



Unsupervised Data Augmentation (UDA) for Consistency Training, Xie et al., NeurIps 2019

$$\min_{\mathbf{w}} \overbrace{\mathbf{E}_{\mathbf{x},y\in L}[-\log p_{\mathbf{w}}(y\,|\,\mathbf{x})]}^{\text{cross entropy}} + \lambda \overbrace{\mathcal{D}_{KL}\left(p_{\mathbf{w}}(y\,|\,\mathbf{x})\,||\,p_{\mathbf{w}}(y\,|\,\hat{\mathbf{x}})\right)}^{\text{consistency in augmentation}}$$

$$E[g] = \sum p(g) \cdot g$$ definition of expected value

$$E[-\log p_{\mathbf{w}}(y\,|\,\mathbf{x})] = -\sum p(y) \cdot \log p_{\mathbf{w}}(y\,|\,\mathbf{x})$$ insert -log probability, log likelihood

$$NLL(y, p_{\mathbf{w}}(y\,|\,\mathbf{x})) = -\sum_c p(y=c) \cdot \log p_{\mathbf{w}}(y=c\,|\,\mathbf{x})$$ negative log likelihood

$$CE(f,g) = -\sum f(x) \cdot \log g(x)$$ cross entropy of two functions

$$CE(y, p_{\mathbf{w}}(y\,|\,\mathbf{x})) = -\sum_c (y=c) \cdot \log p_{\mathbf{w}}(y=c\,|\,\mathbf{x})$$ if y=c is a probability, these are same equation

```
cce = tf.keras.losses.CategoricalCrossentropy()
cce(y_true, y_pred)
```