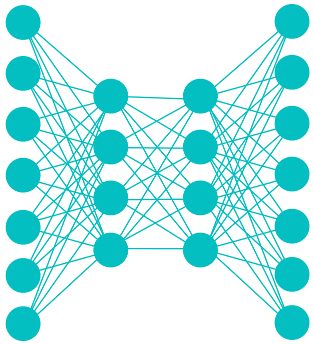
Lecture Notes for

Neural Networks and Machine Learning

Self-supervised, Multi-modal, & Multi-task Learning



Logistics and Agenda

- Logistics
  - Newest Lab uses multi-task / multi-modal learning
- Agenda
  - Paper Presentation: X-vectors
  - Finish Self Supervised Learning
  - Multi-modal/task Learning
    - Techniques
    - Applications and domains
- Next Time:
  - Paper Presentation: DeepTox

Paper Presentation: Speaker Embedding

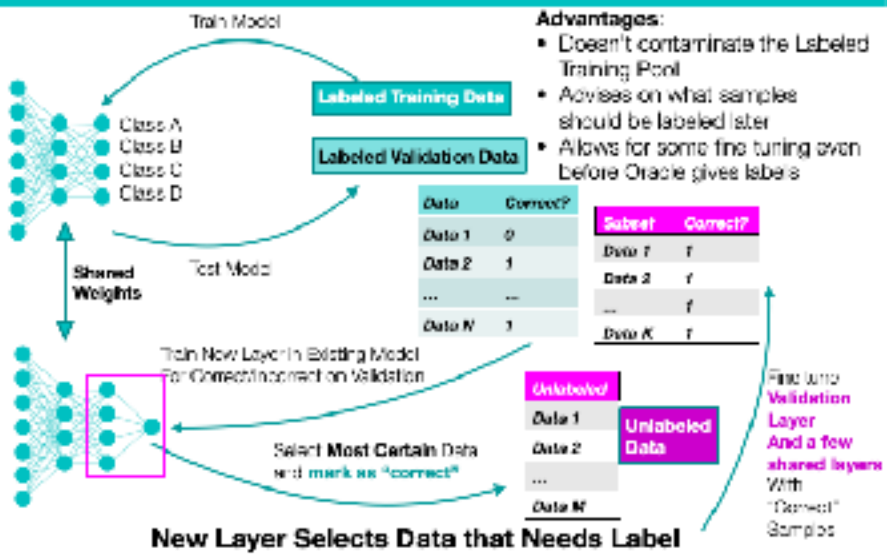
Attentive Statistics Pooling for Deep Speaker Embedding

Koji Okabe[1], *Takafumi Koshinaka*[1], *Koichi Shinoda*[2]

[1]Data Science Research Laboratories, NEC Corporation, Japan
[2]Department of Computer Science, Tokyo Institute of Technology, Japan
k-okabe@bx.jp.nec.com, koshinak@ap.jp.nec.com, shinoda@c.titech.ac.jp

Last Time

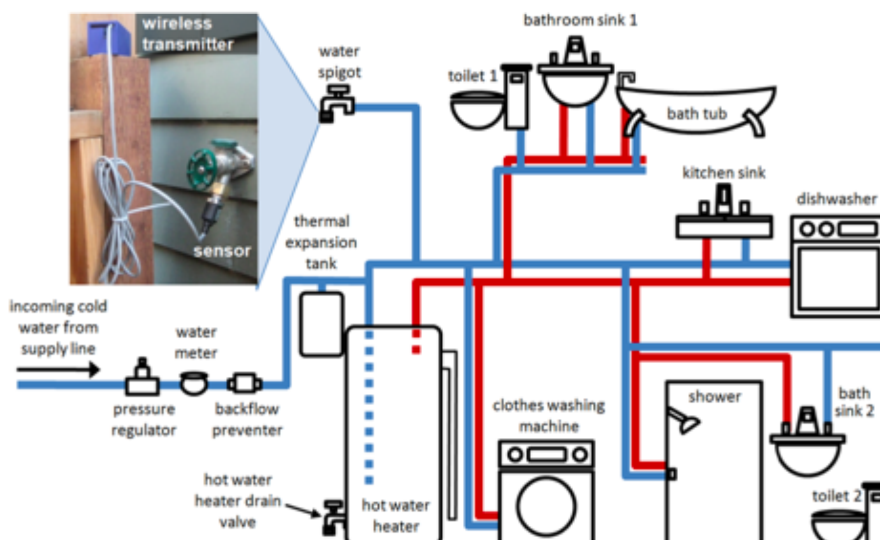
ATLAS: Active Transfer Learning for Adaptive Sampling



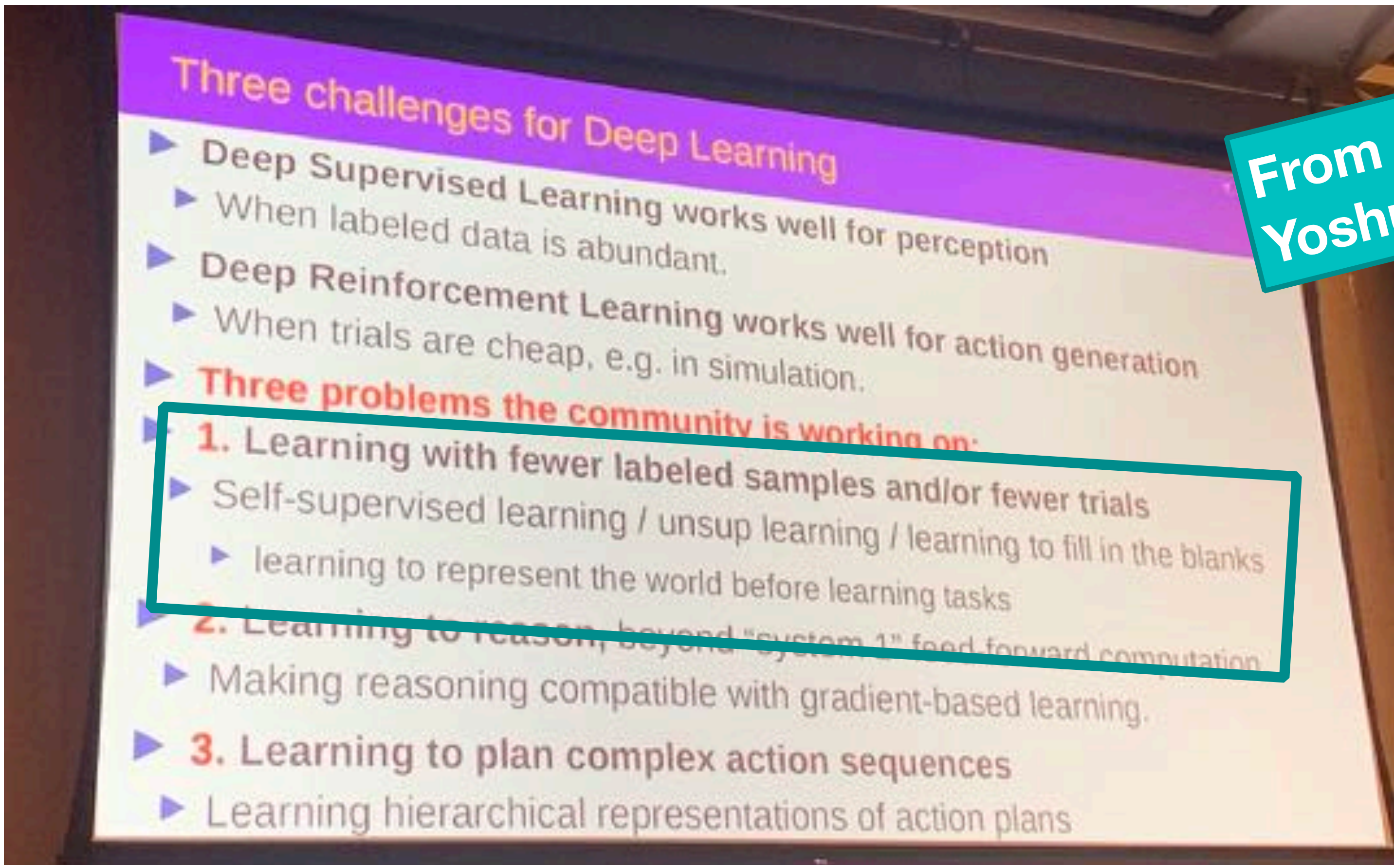
Started: Self Supervised Learning
Auxiliary tasks to learn about the world

Active Learning Overview

- **Basic Idea**: Use a trained model to sample from an oracle that can magically give you a new label
  - Active Learning:
    What labels should we ask the oracle about?
- Uncertainty Sampling
  - Choose instances where the model is most uncertain or most certain
  - Various ways to measure certainty
- Diversity Sampling
  - Choose instances that are similar or different from training distribution



Self-Supervised Learning

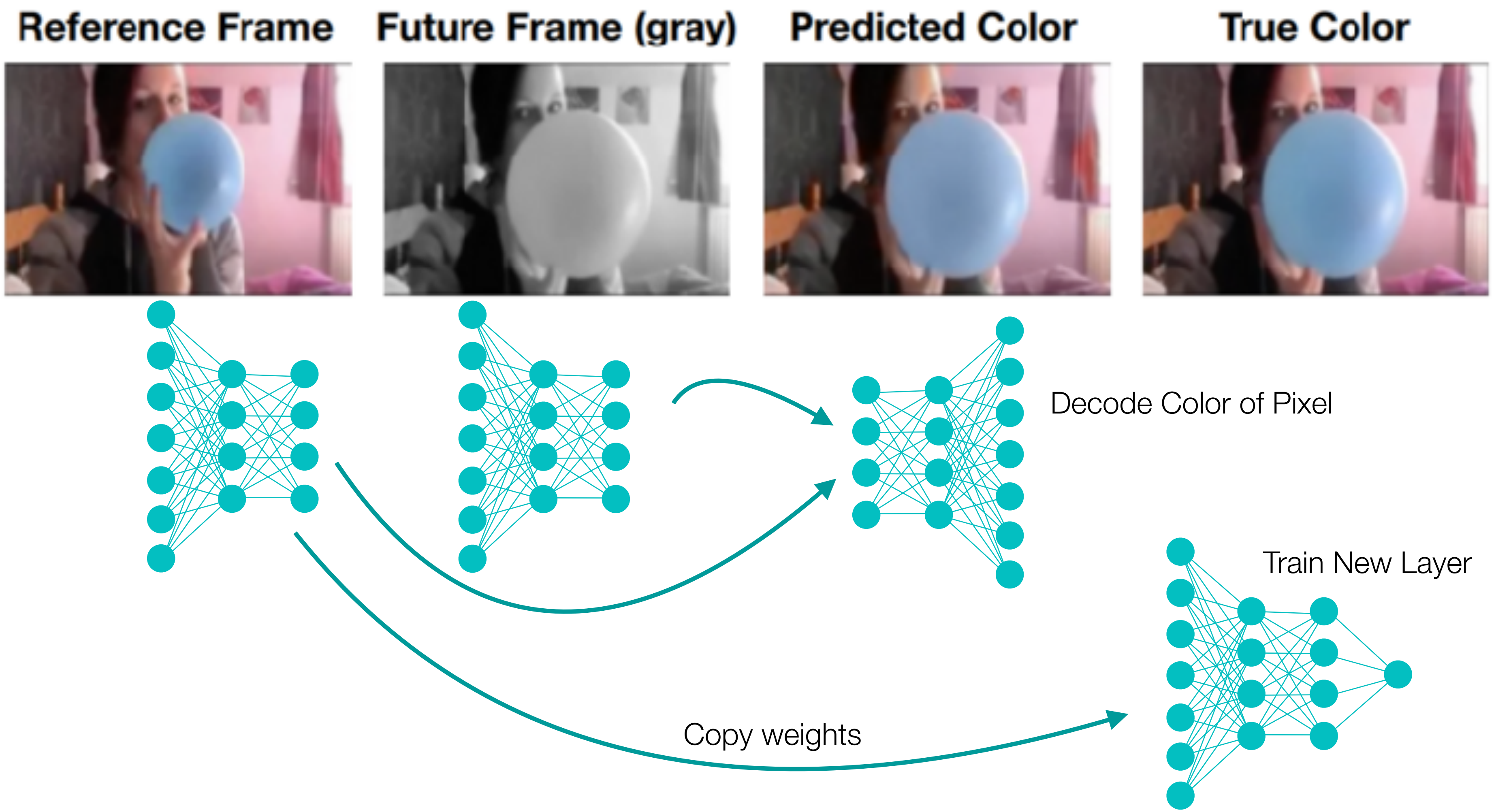


From Yoshua Bengio

Self-supervised Learning

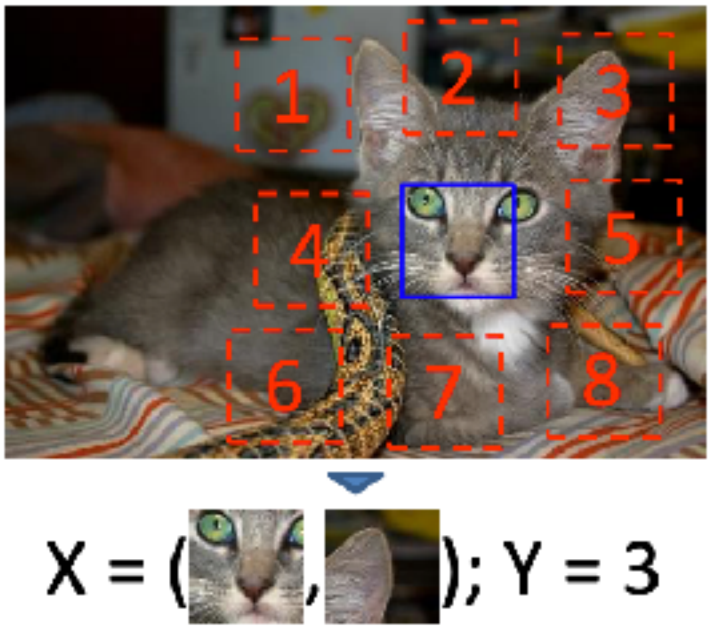
- **Problem**: deep learning is not sample efficient
- **Idea**: learn about the world before learning the task
- **New Problem**: how do we learn about the world?
- **Solution**: transfer learning on toy problem
  - 1. train on auxiliary task that is easy to label
  - 2. throw away anything specific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

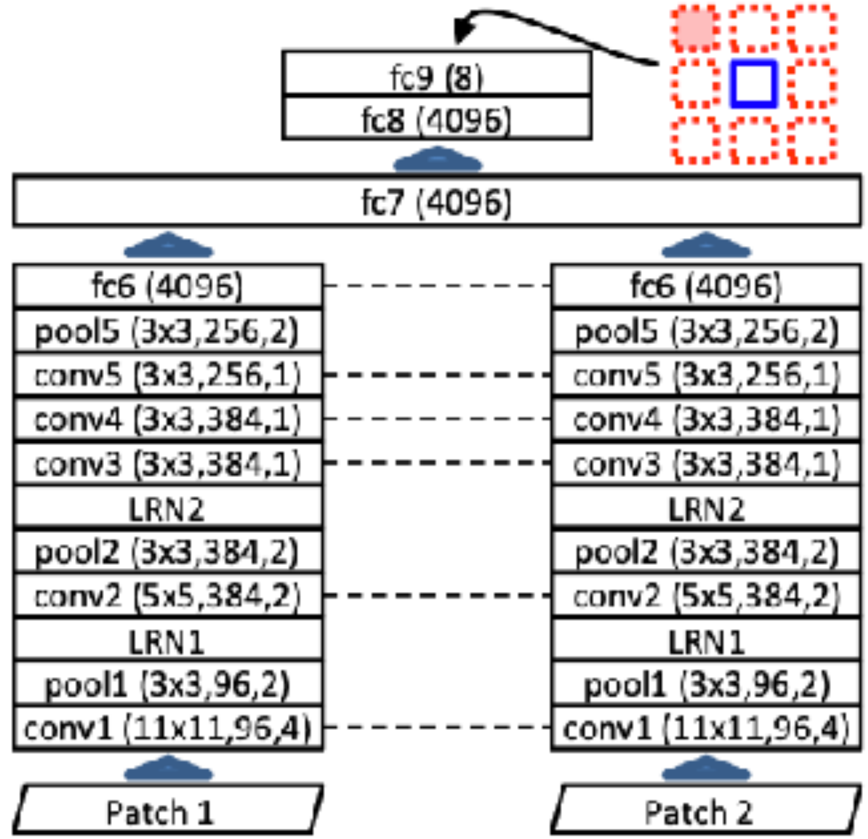
Examples of Self Supervised Learning



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 SSL

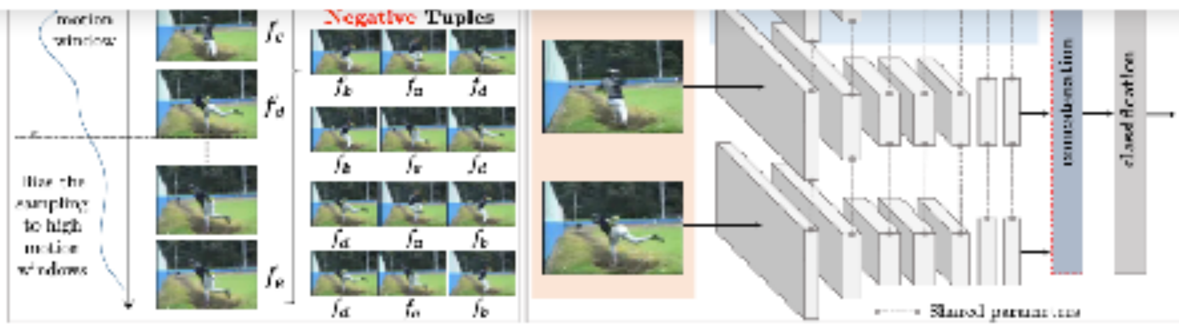
Shuffle and Learn: Unsupervised Learning using Temporal Order Verification

Ishan Misra[1] C. Lawrence Zitnick[2] Martial Hebert[1]

[1] The Robotics Institute, Carnegie Mellon University
[2] Facebook AI Research

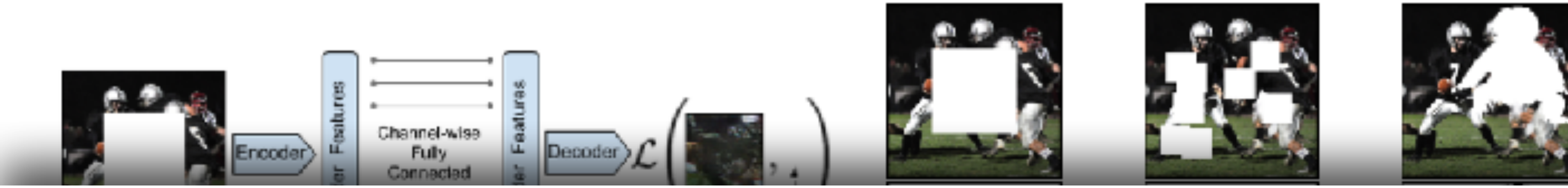
Table 2: Mean classification accuracies over the 3 splits of UCF101 and HMDB51 datasets. We compare different initializations and finetune them for action recognition.

Dataset	Initialization	Mean Accuracy
UCF101	Random	38.6
	(Ours) Tuple verification	**50.2**
HMDB51	Random	13.3
	UCF Supervised	15.2
	(Ours) Tuple verification	**18.1**

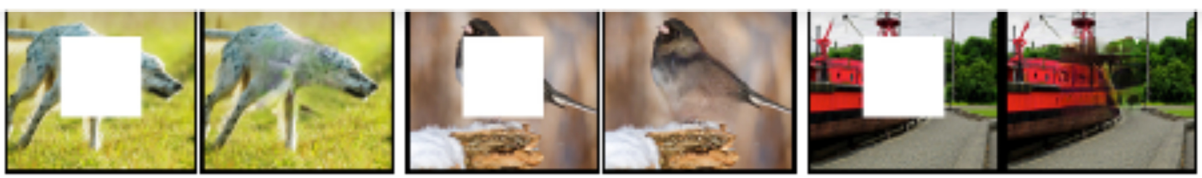


https://www.fast.ai/2020/01/13/self_supervised/

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

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

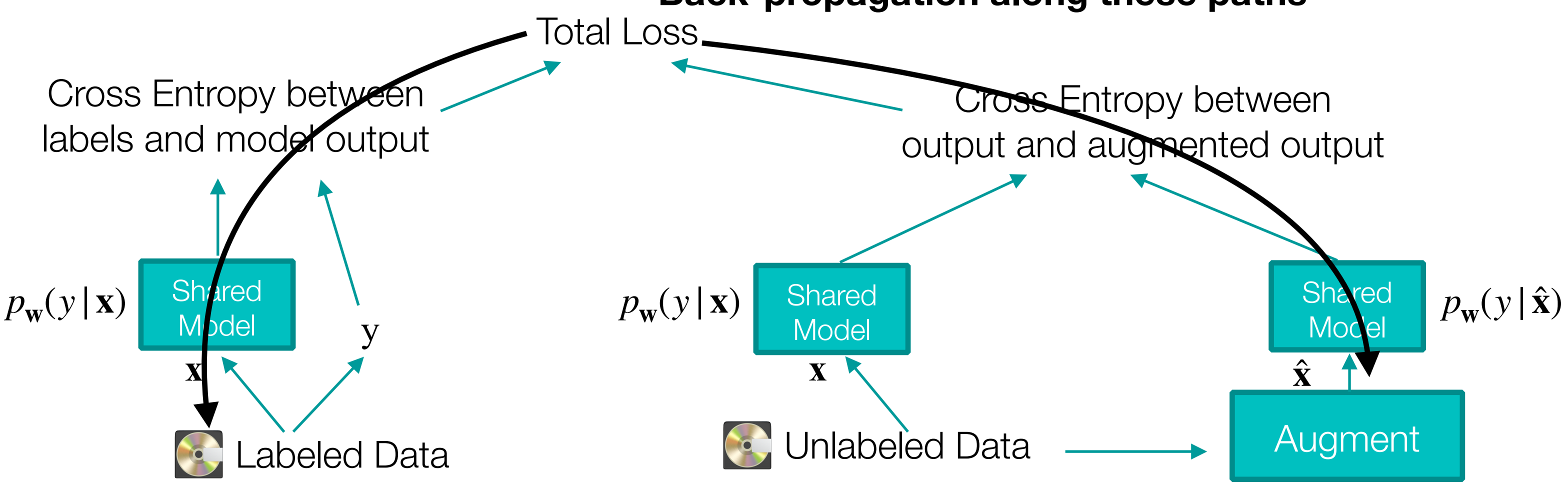
no back prop **yes** back prop

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

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

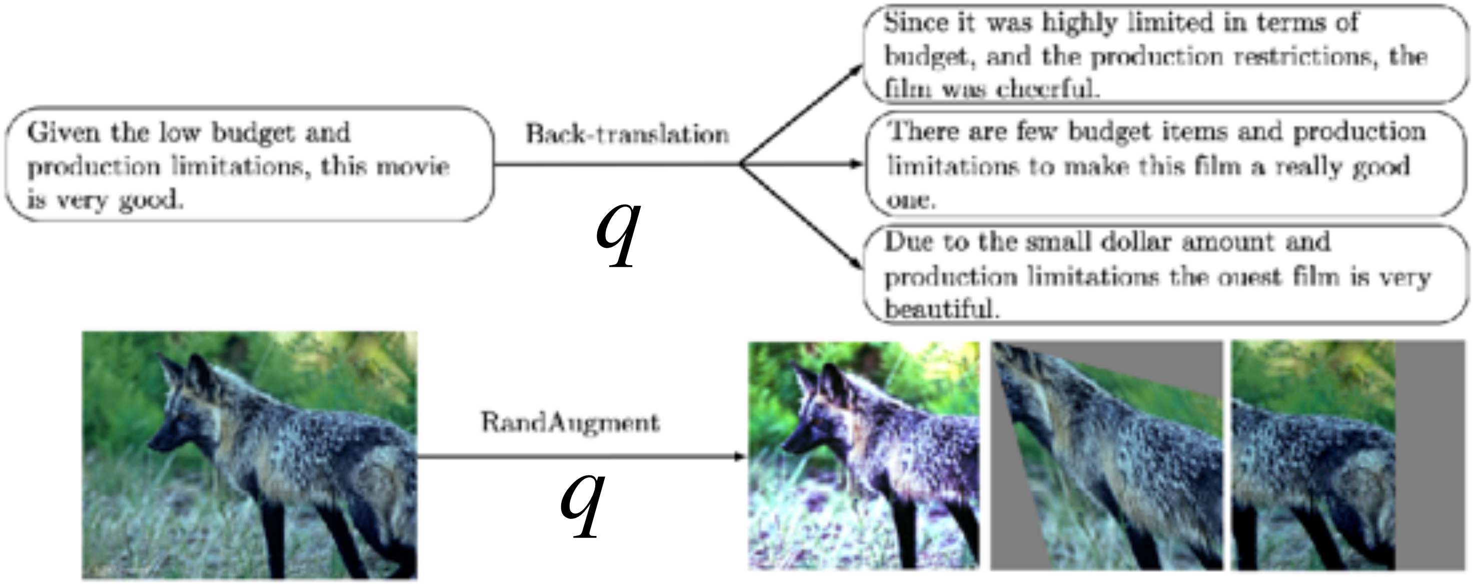


Figure 2: Augmented examples using back-translation and RandAugment.

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

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

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

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

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

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

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

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

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

$$\mathcal{D}_{KL}(f\,||\,g) = -\sum f(x)\cdot\log\frac{g(x)}{f(x)}$$ definition of Kullback-Leibler (KL) Divergence

$$\mathcal{D}_{KL}(p(y\,|\,\mathbf{x})\,||\,p(y\,|\,\hat{\mathbf{x}})) = -\sum p(y\,|\,\mathbf{x})\cdot\log\frac{p(y\,|\,\hat{\mathbf{x}})}{p(y\,|\,\mathbf{x})} = -\sum p(y\,|\,\mathbf{x})\cdot\left(\log p(y\,|\,\hat{\mathbf{x}}) - \log p(y\,|\,\mathbf{x})\right)$$

$$= -\sum p(y\,|\,\mathbf{x})\cdot\log p(y\,|\,\hat{\mathbf{x}}) + \sum p(y\,|\,\mathbf{x})\cdot\log p(y\,|\,\mathbf{x})$$

$$= \mathbf{E}_{\mathbf{x}\in U,\hat{\mathbf{x}}\leftarrow q(\hat{\mathbf{x}}|\mathbf{x})}\left[-\log p(y\,|\,\hat{\mathbf{x}})\right] + \mathbf{E}_{\mathbf{x}\in U}\left[\cancel{\log p(y\,|\,\mathbf{x})}\right]$$ ignore

cross entropy of unsupervised labels after augmentation

entropy of unsupervised labels ***constant***

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

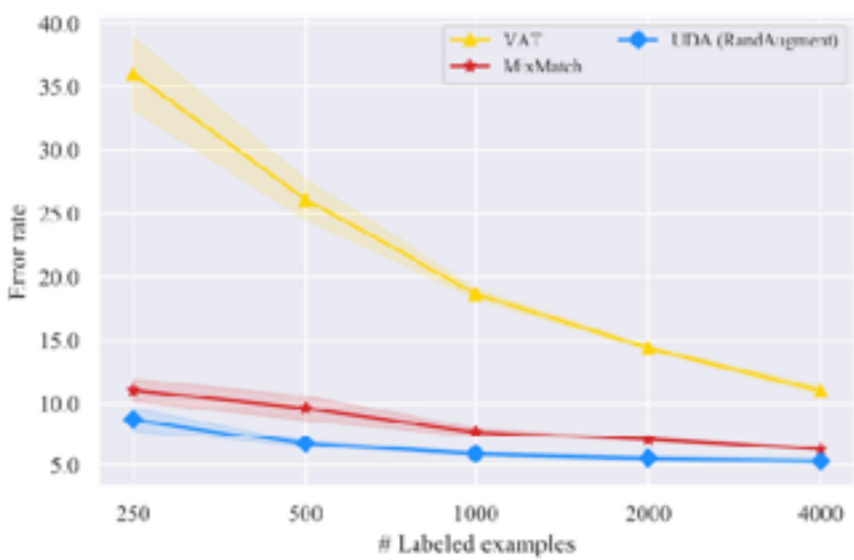
Unsupervised Consistency Loss

Augmentation (# Sup examples)	Sup (50k)	Semi-Sup (4k)
Crop & flip	5.36	16.17
Cutout	4.42	6.42
RandAugment	**4.23**	**5.29**

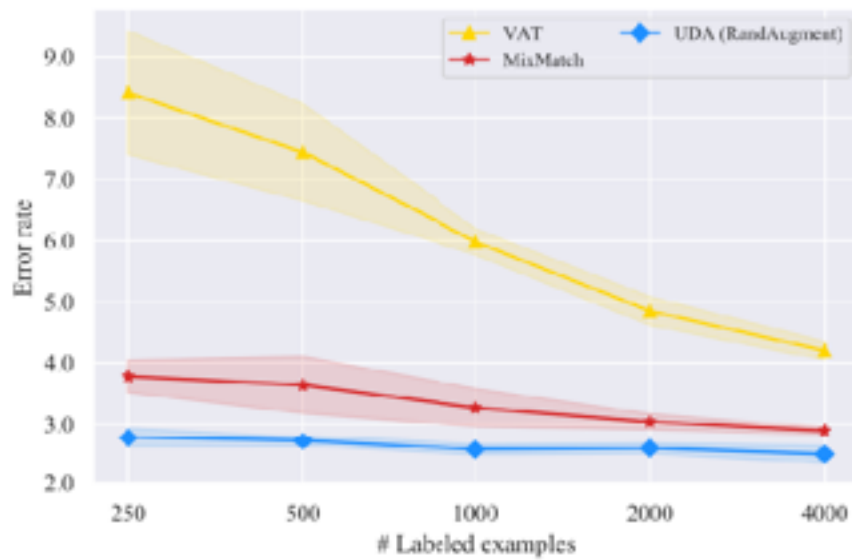
Table 1: Error rates on CIFAR-10.

Augmentation (# Sup examples)	Sup (650k)	Semi-sup (2.5k)
✗	38.36	50.80
Switchout	37.24	43.38
Back-translation	**36.71**	**41.35**

Table 2: Error rate on Yelp-5.



(a) CIFAR-10



(b) SVHN

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

Unsupervised Consistency Loss

Method	Model	# Param	CIFAR-10 (4k)	SVHN (1k)
Π-Model (Laine & Aila, 2016)	Conv-Large	3.1M	12.36 ± 0.31	4.82 ± 0.17
Mean Teacher (Tarvainen & Valpola, 2017)	Conv-Large	3.1M	12.31 ± 0.28	3.95 ± 0.19
VAT + EntMin (Miyato et al., 2018)	Conv-Large	3.1M	10.55 ± 0.05	3.86 ± 0.11
SNTG (Luo et al., 2018)	Conv-Large	3.1M	10.93 ± 0.14	3.86 ± 0.27
VAdD (Park et al., 2018)	Conv-Large	3.1M	11.32 ± 0.11	4.16 ± 0.08
Fast-SWA (Athiwaratkun et al., 2018)	Conv-Large	3.1M	9.05	-
ICT (Verma et al., 2019)	Conv-Large	3.1M	7.29 ± 0.02	3.89 ± 0.04
Pseudo-Label (Lee, 2013)	WRN-28-2	1.5M	16.21 ± 0.11	7.62 ± 0.29
LGA + VAT (Jackson & Schulman, 2019)	WRN-28-2	1.5M	12.06 ± 0.19	6.58 ± 0.36
mixmixup (Hataya & Nakayama, 2019)	WRN-28-2	1.5M	10	-
ICT (Verma et al., 2019)	WRN-28-2	1.5M	7.66 ± 0.17	3.53 ± 0.07
MixMatch (Berthelot et al., 2019)	WRN-28-2	1.5M	6.24 ± 0.06	2.89 ± 0.06

Methods	SSL	10%	100%
ResNet-50	✗	55.09 / 77.26	77.28 / 93.73
w. RandAugment	✗	58.84 / 80.56	78.43 / 94.37
UDA (RandAugment)	✓	**68.78 / 88.80**	**79.05 / 94.49**

Table 5: Top-1 / top-5 accuracy on ImageNet with 10% and 100% of the labeled set. We use image size 224 and 331 for the 10% and 100% experiments respectively.

Lecture Notes for

Neural Networks and Machine Learning

Multi-Modal and Multi-Task

Next Time:
Demo

Reading: Papers