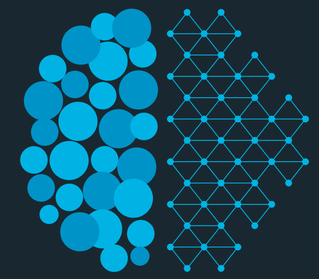



Institut
des algorithmes
d'apprentissage
de Montréal

# Introduction to IFT6268
# Self-supervised Representation Learning

Aaron Courville
Université de Montréal

IFT6268 - Self-Supervised Representation Learning

Slides and slide material from Hugo Larochelle, Devon Hjelm, Samuel Lavoie and Faruk Ahmed.

# SSL in the media

**Self-supervised learning is the future of AI**

The Paradigm Shift of Self-Supervised Learning

Carlos E. Perez Follow

May 23, 2019 · 6 min read


Photo by Nicolas Cool on Unsplash

*“ If I’m just trying to predict what happens next, that’s supervised learning because what happens next acts as the label, but I don ’t need to add extra labels. There’s this thing in between unlabeled data and labeled data, which is predicting what comes next.”* — **Geoffrey Hinton** (in an interview in “Architects of Intelligence”)

# Learning “really useful” representations


- Longstanding dream of the Deep Learning community:
  - Use unsupervised learning to learn some feature representation that can be used to support effective supervised learning (like classification)

**Input**

**Extract Information**

Representation

**Tasks**

Human-level interaction

Dialogue

Generation

Translation

Anomaly detection

Retrieval

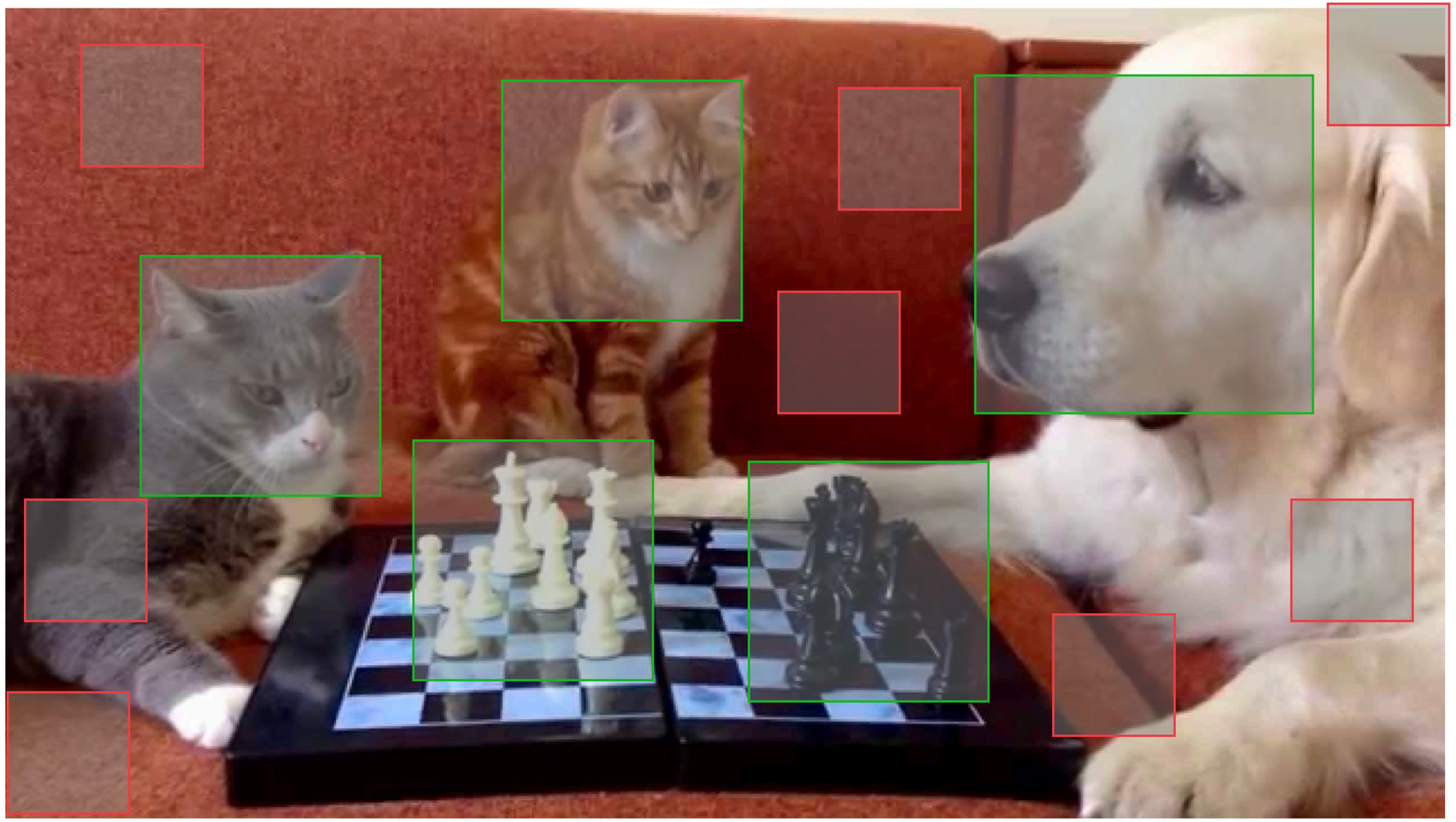
Semantic Segmentation

Classification

**More utility = better**

**(Task) Generalization <—> Understanding**

# We don’t need generation/reconstruction



**Interesting thing**

**Not interesting thing**

*Generative models (in principle) care about all the pixels*

# Representation Learning for Supervised Learning


- Using generative models (AEs, VAEs, etc) have largely been ineffective with two caveats:
    1. Natural Language Modelling (all SOTA models are build on BERT-like representations)
    2. In the very small dataset regime, unsupervised learning can actually help.
- Gradient-based supervised training with the right model (eg. CNNs for vision problems) has been very difficult to beat with unsupervised methods.

# It's worth asking … Why?

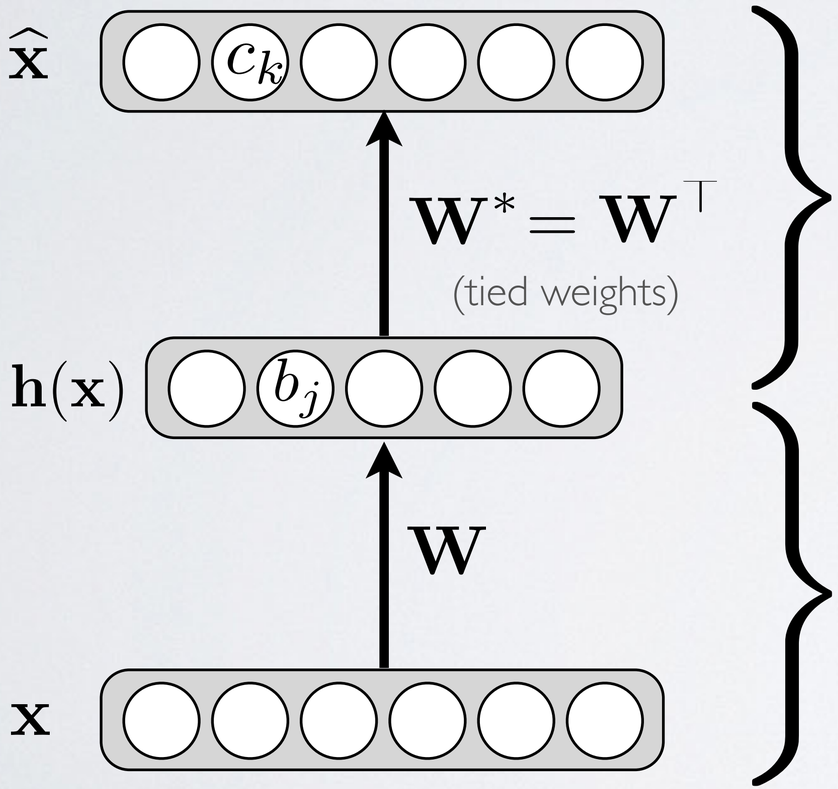

A speculative answer:

- Most (essentially all) existing unsupervised methods learn features that are overwhelmingly low-level (nonsemantic).
  - The features describe superficial aspects of the data and preserve few of the invariances that one would want from a representation learning scheme.
- Modern supervised learning methods (i.e. with NN) learn layers of representations that learn the relevant axes of variance in the data.
  - Eg. Higher level features of a CNN trained to recognize car makes and models  should be relatively invariance to color but very sensitive to subtle differences in shape.

# AUTOENCODER

**Topics:** autoencoder, encoder, decoder, tied weights

- Feed-forward neural network trained to reproduce its input at the output layer




Decoder

$$\begin{aligned}\hat{\mathbf{x}} &= o(\hat{\mathbf{a}}(\mathbf{x})) \\ &= \underbrace{\mathrm{sigm}}_{\text{for binary inputs}}(\mathbf{c} + \mathbf{W}^*\mathbf{h}(\mathbf{x}))\end{aligned}$$

Encoder

$$\begin{aligned}\mathbf{h}(\mathbf{x}) &= g(\mathbf{a}(\mathbf{x})) \\ &= \mathrm{sigm}(\mathbf{b} + \mathbf{W}\mathbf{x})\end{aligned}$$

# AUTOENCODER

**Topics:** loss function

- For binary inputs:

$$f(\mathbf{x}) \equiv \hat{\mathbf{x}}$$

$$l(f(\mathbf{x})) = -\sum_k \left(x_k \log(\hat{x}_k) + (1 - x_k) \log(1 - \hat{x}_k)\right)$$

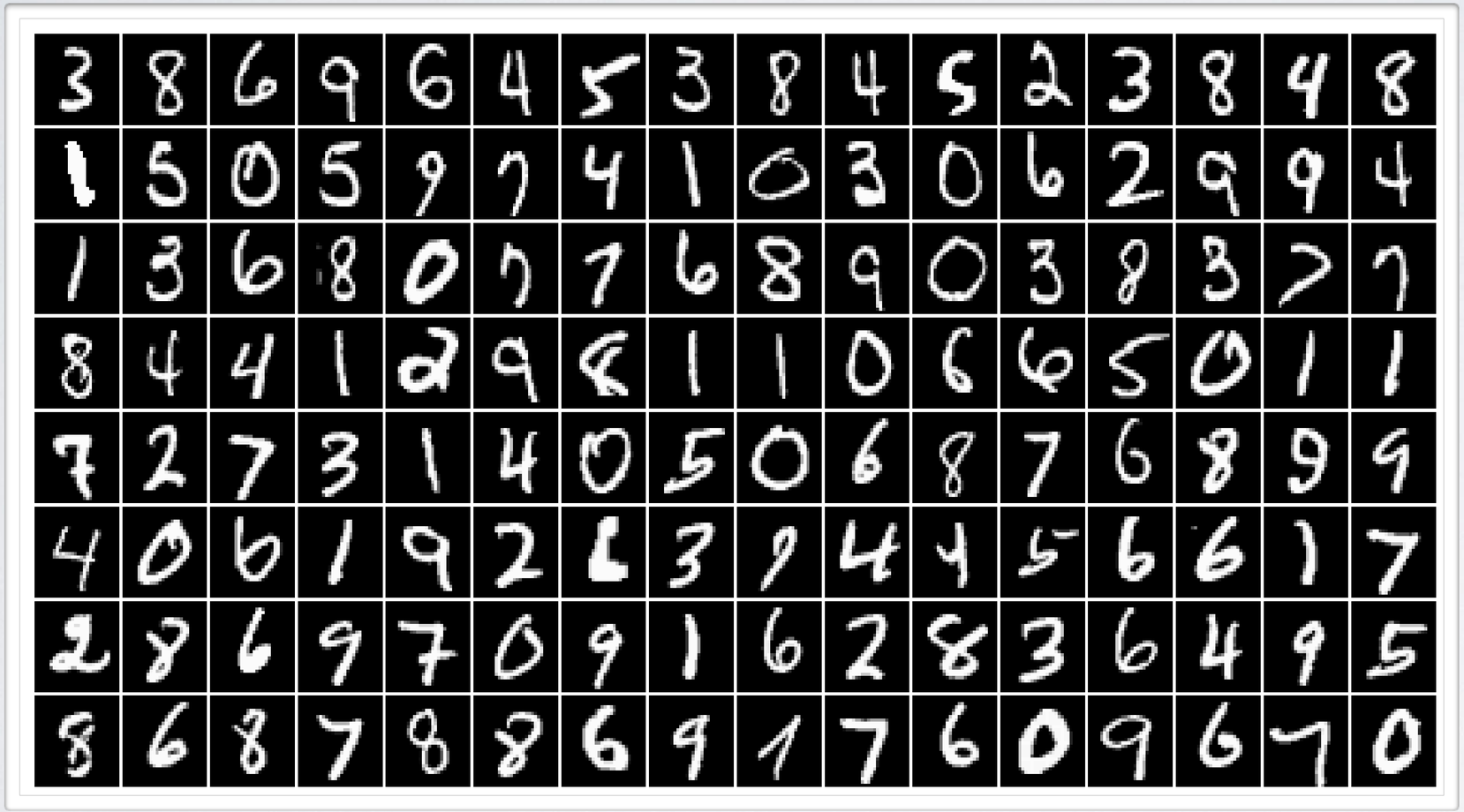
  - cross-entropy (more precisely: sum of Bernoulli cross-entropies)

- For real-valued inputs:

$$l(f(\mathbf{x})) = \frac{1}{2} \sum_k (\hat{x}_k - x_k)^2$$

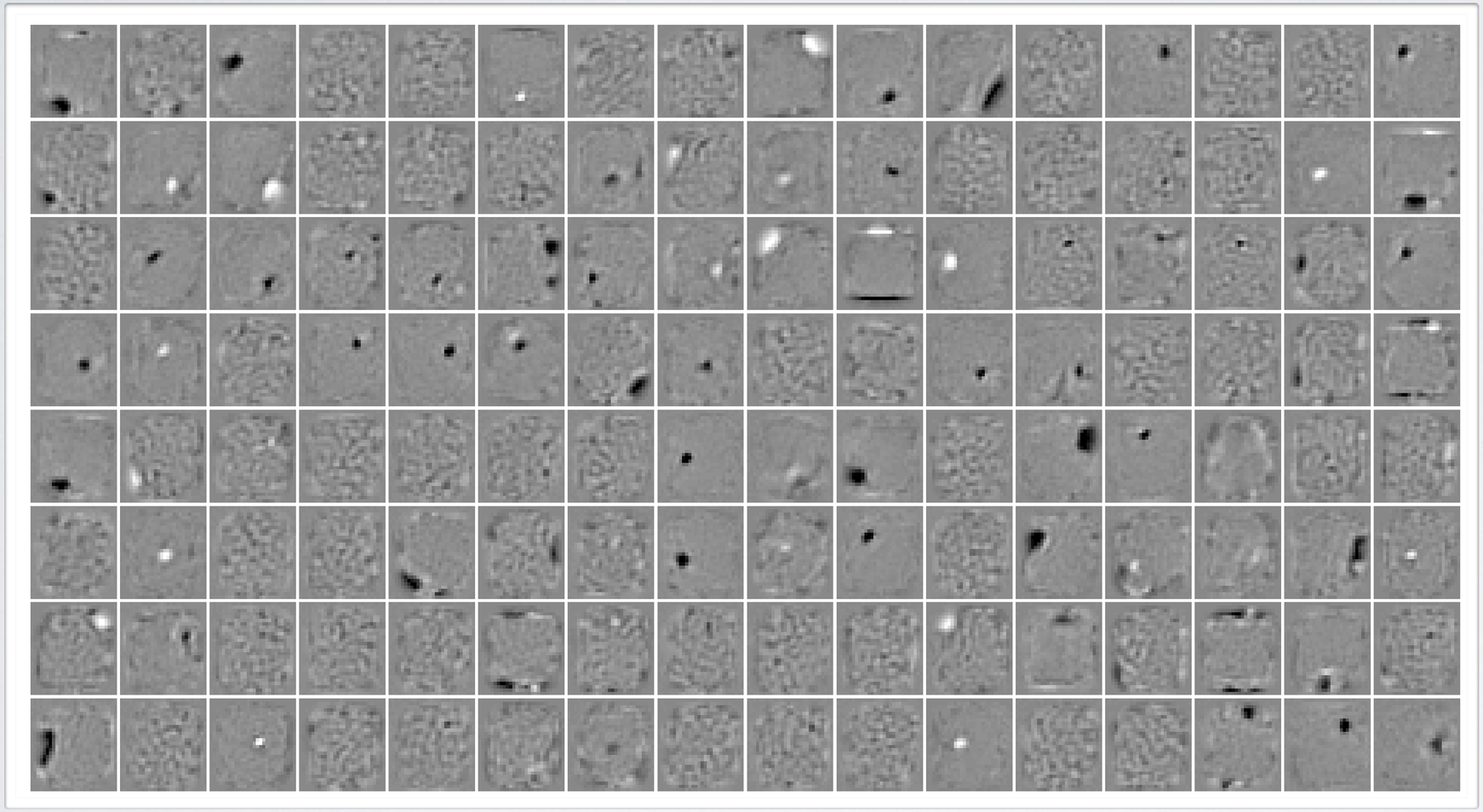
  - sum of squared differences (squared euclidean distance)
  - we use a linear activation function at the output

# EXAMPLE OF DATA SET: MNIST



# FILTERS (AUTOENCODER)

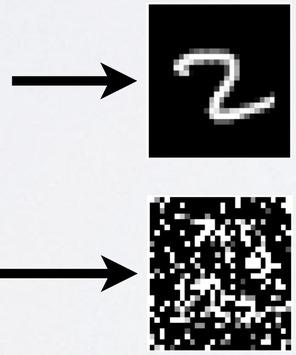
(Larochelle et al., JMLR2009)



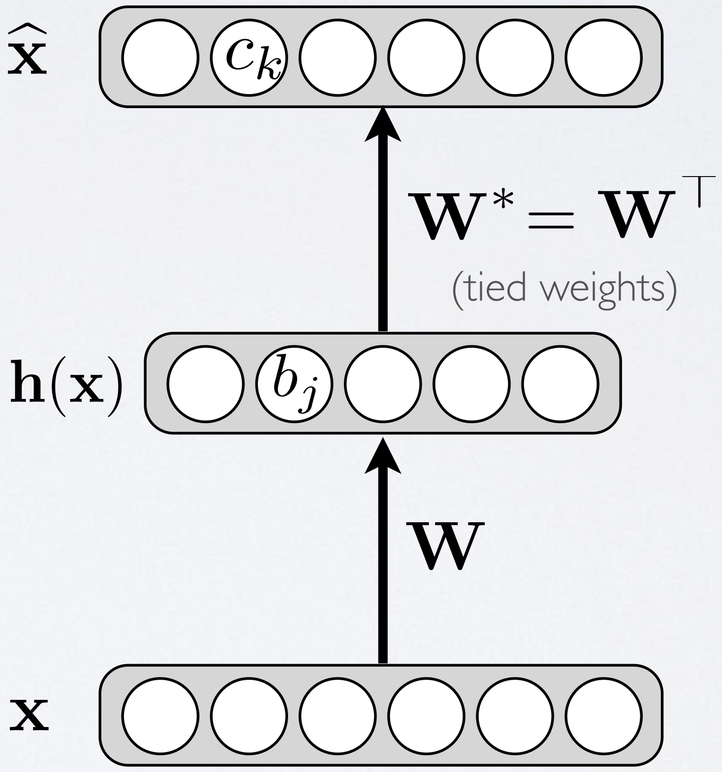
# UNDERCOMPLETE HIDDEN LAYER

**Topics:** undercomplete representation

- Hidden layer is undercomplete if smaller than the input layer
  - hidden layer "compresses" the input
  - will compress well only for the training distribution
- Hidden units will be
  - good features for the training distribution
  - but bad for other types of input



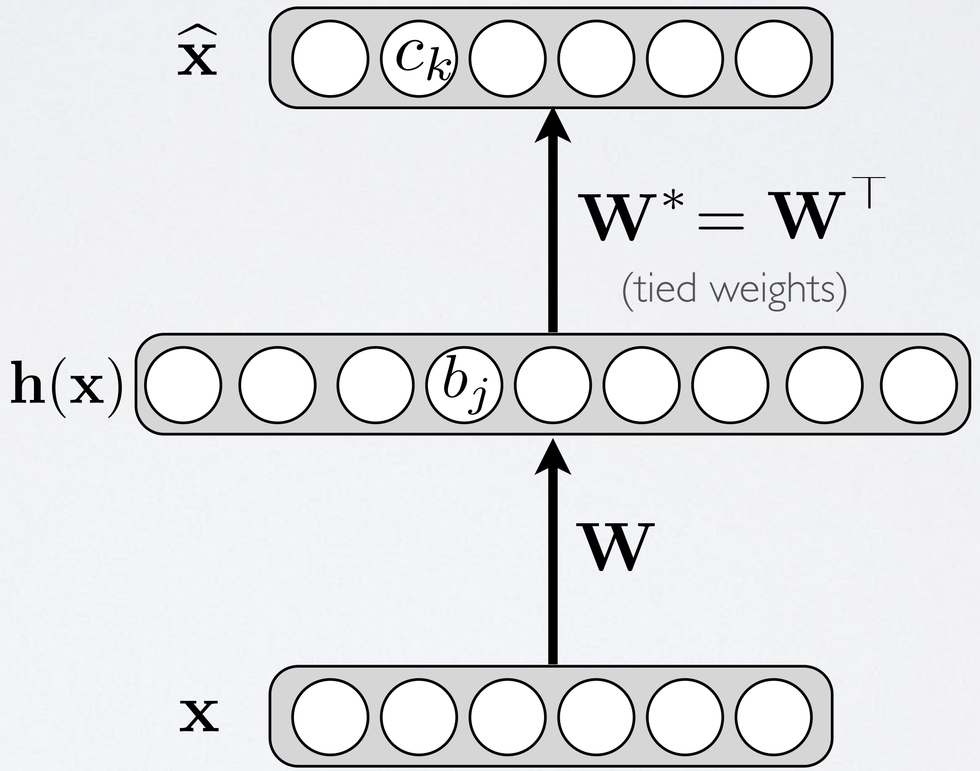




# OVERCOMPLETE HIDDEN LAYER

**Topics:** overcomplete representation

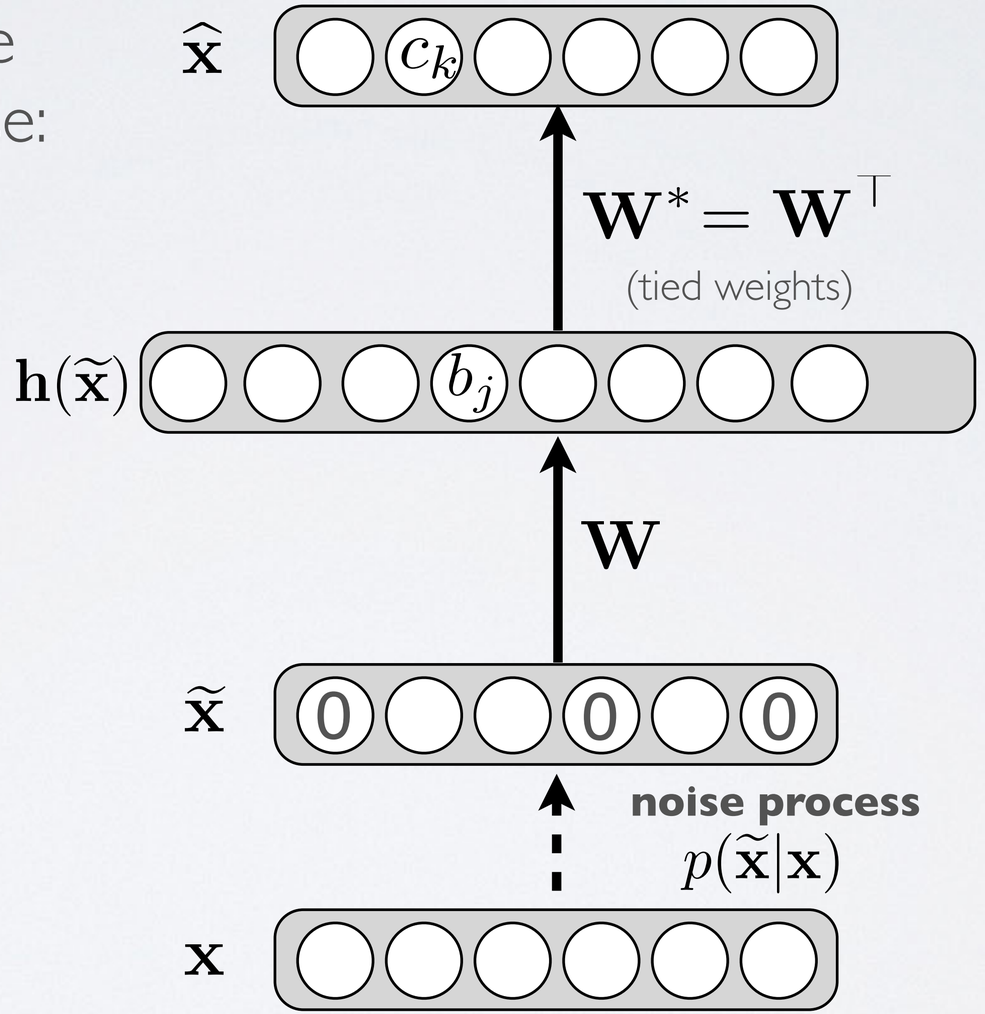
- Hidden layer is overcomplete if greater than the input layer
  - no compression in hidden layer
  - each hidden unit could copy a different input component
- No guarantee that the hidden units will extract meaningful structure



# DENOISING AUTOENCODER

**Topics:** denoising autoencoder

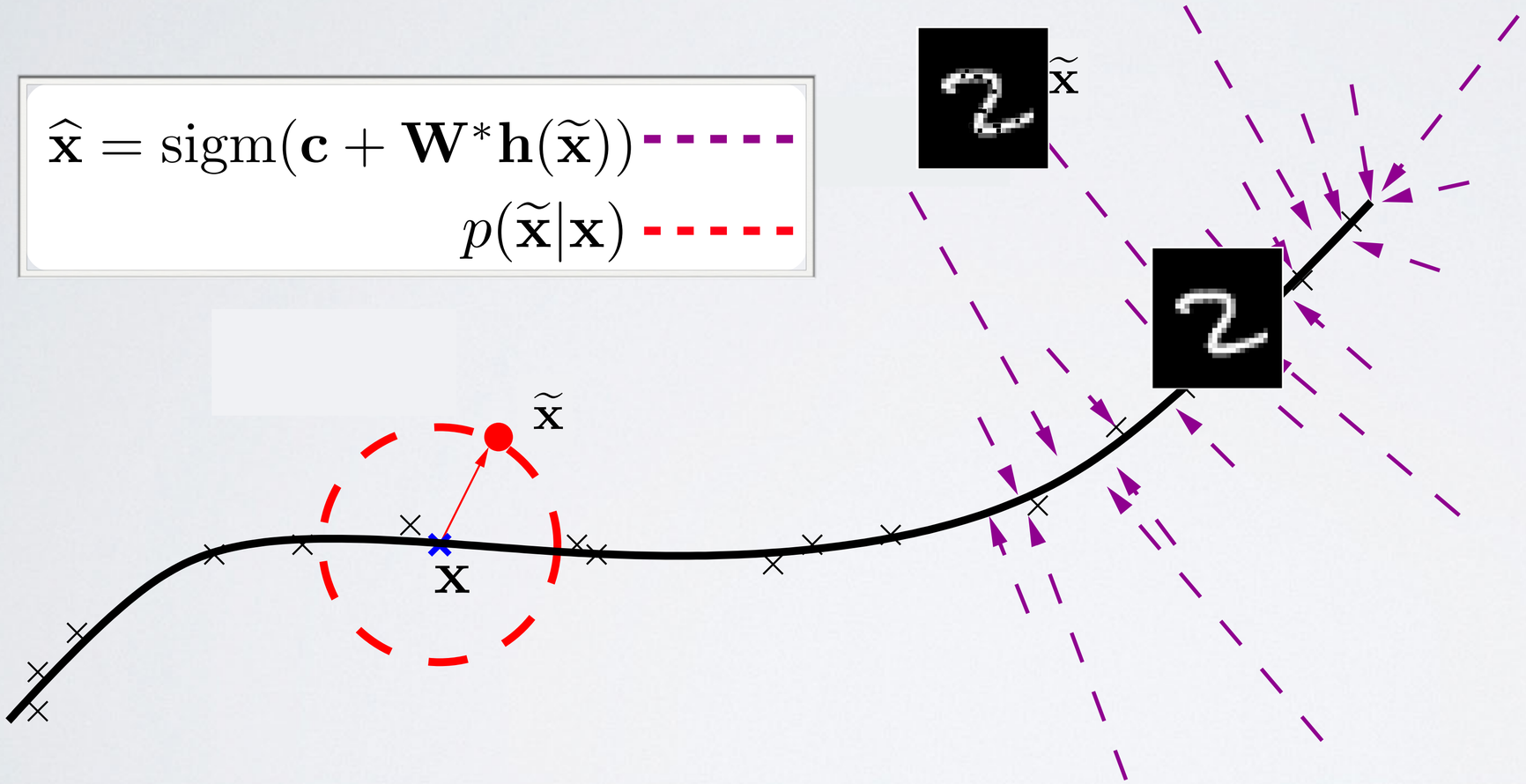
- Idea: representation should be robust to introduction of noise:
  - random assignment of subset of inputs to 0, with probability $\nu$
  - Gaussian additive noise
- Reconstruction $\widehat{\mathbf{x}}$ computed from the corrupted input $\widetilde{\mathbf{x}}$
- Loss function compares $\widehat{\mathbf{x}}$ reconstruction with the **noiseless input** $\mathbf{x}$



# DENOISING AUTOENCODER

$$\widehat{\mathbf{x}} = \mathrm{sigm}(\mathbf{c} + \mathbf{W}^{*}\mathbf{h}(\widetilde{\mathbf{x}}))$$

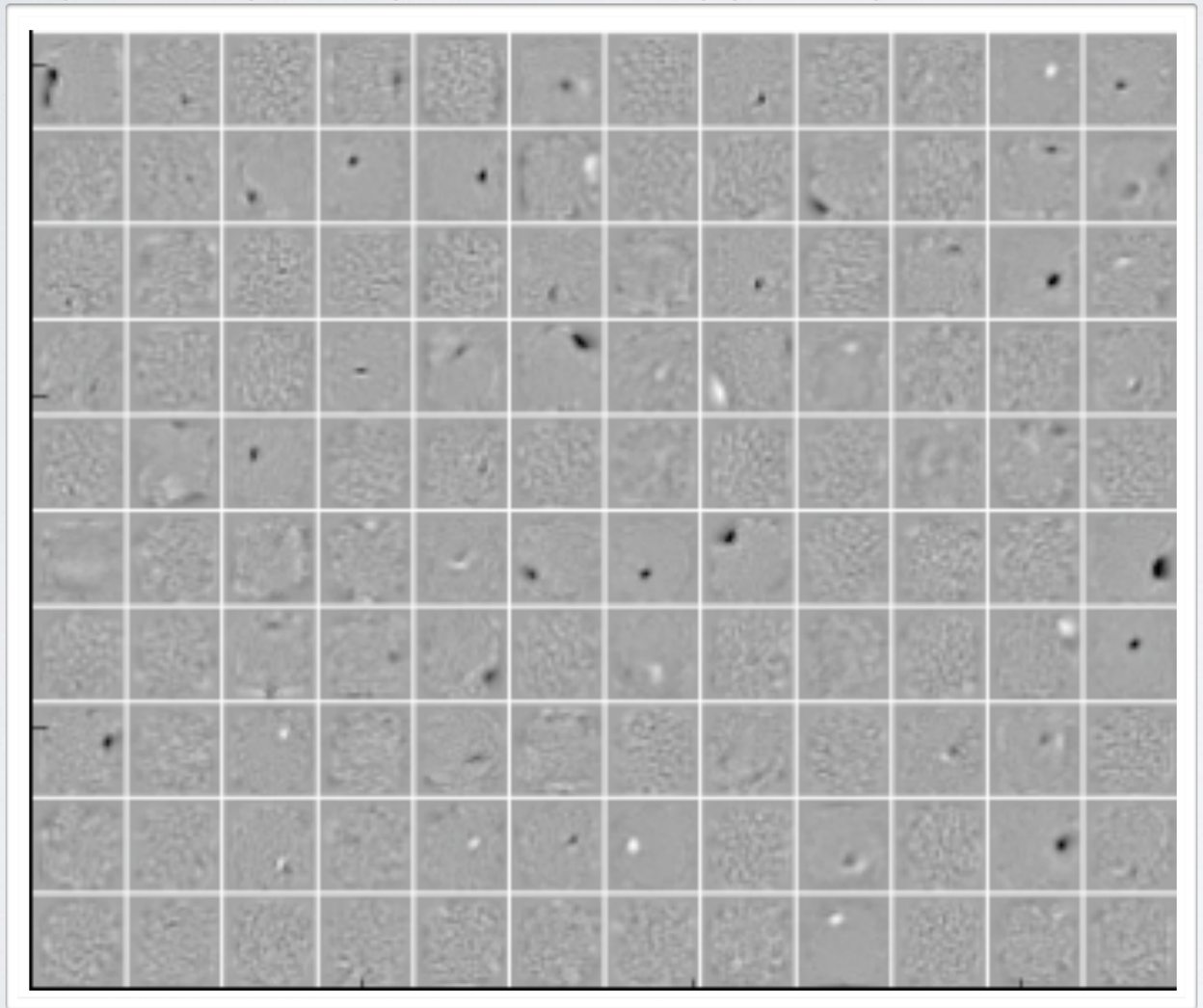
$$p(\widetilde{\mathbf{x}}|\mathbf{x})$$



# FILTERS (DENOISING AUTOENCODER)

(Vincent, Larochelle, Bengio and Manzagol, ICML 2008)

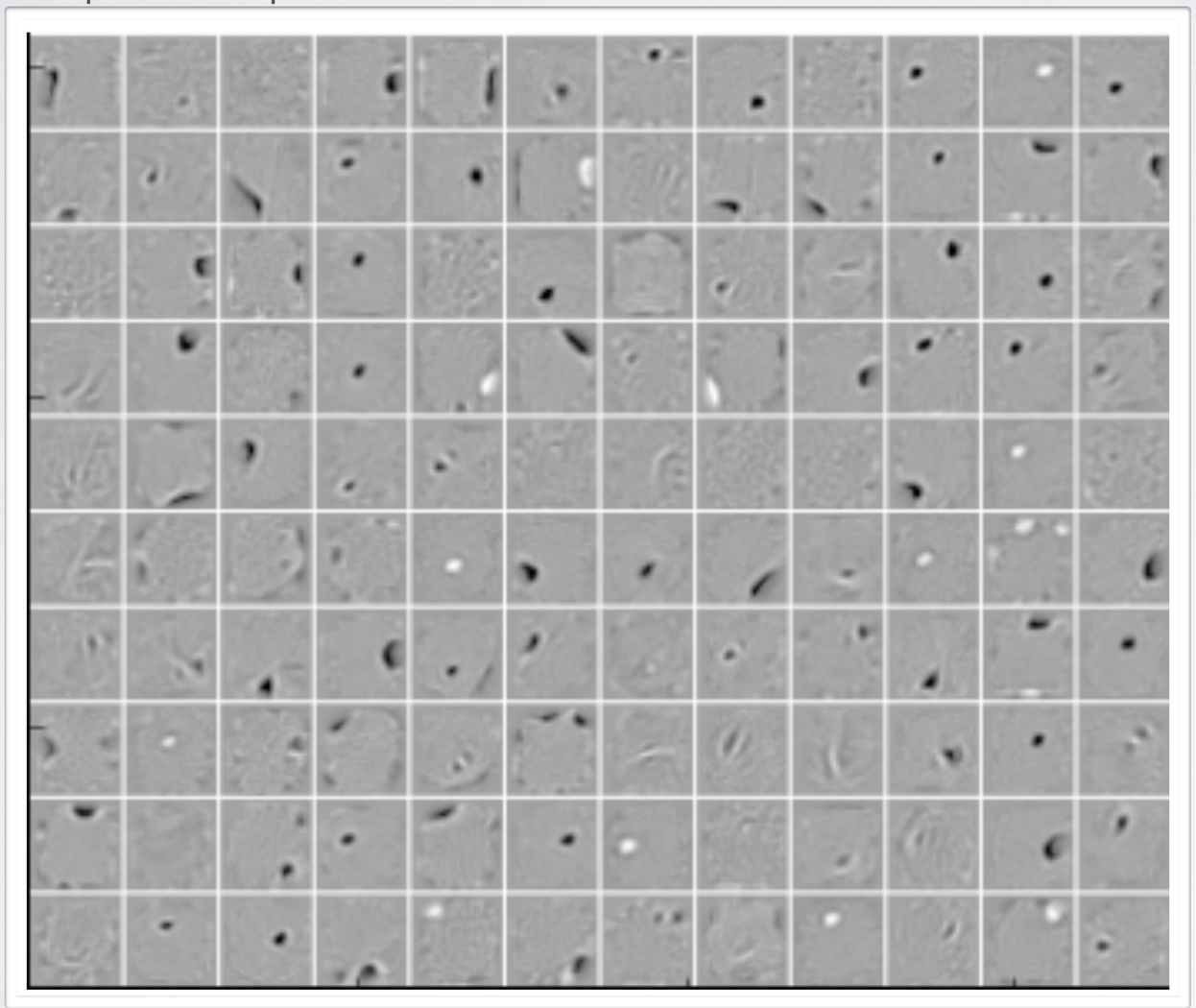
- No corrupted inputs (cross-entropy loss)



# FILTERS (DENOISING AUTOENCODER)

(Vincent, Larochelle, Bengio and Manzagol, ICML 2008)

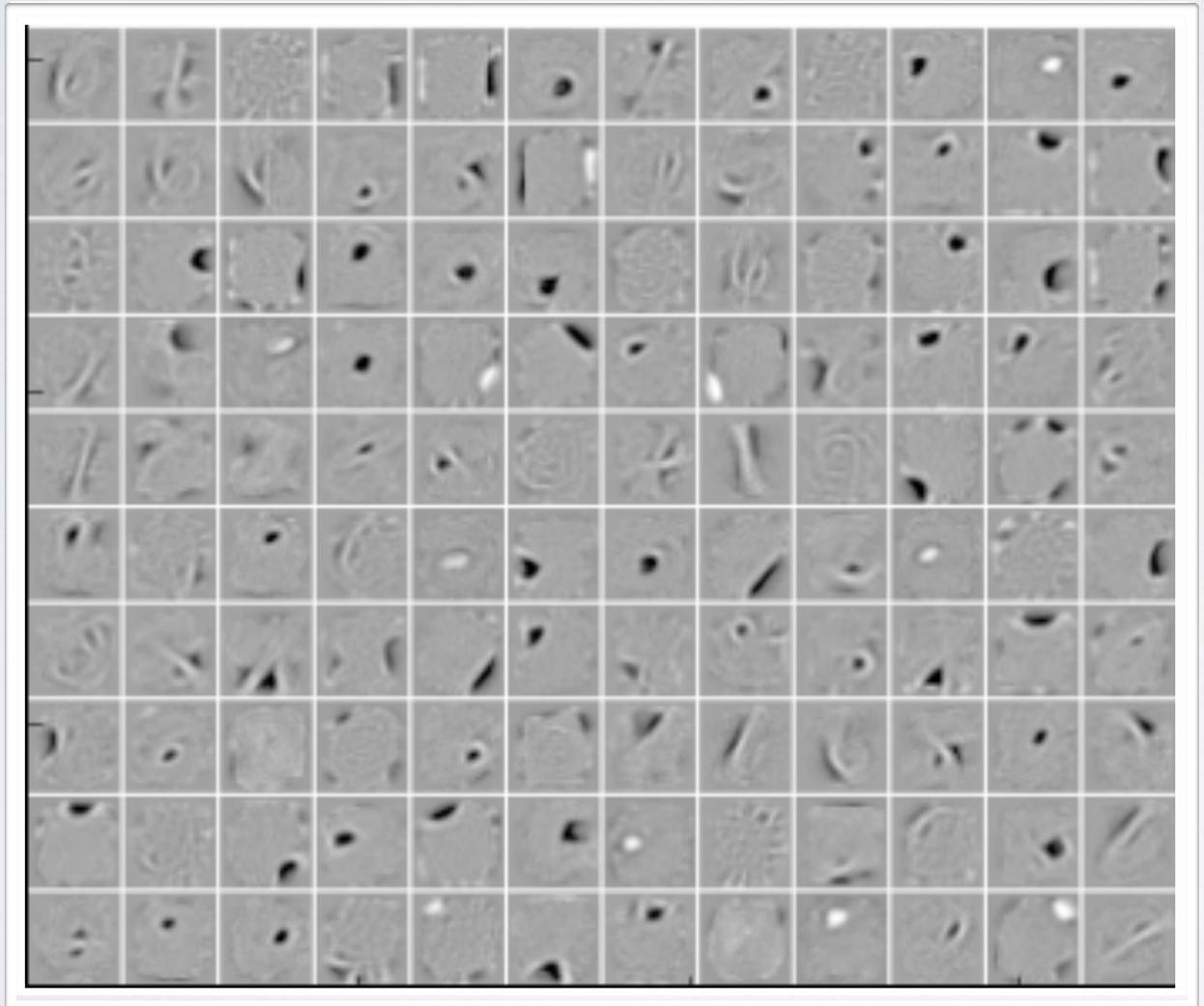
- 25% corrupted inputs



# FILTERS (DENOISING AUTOENCODER)

(Vincent, Larochelle, Bengio and Manzagol, ICML 2008)

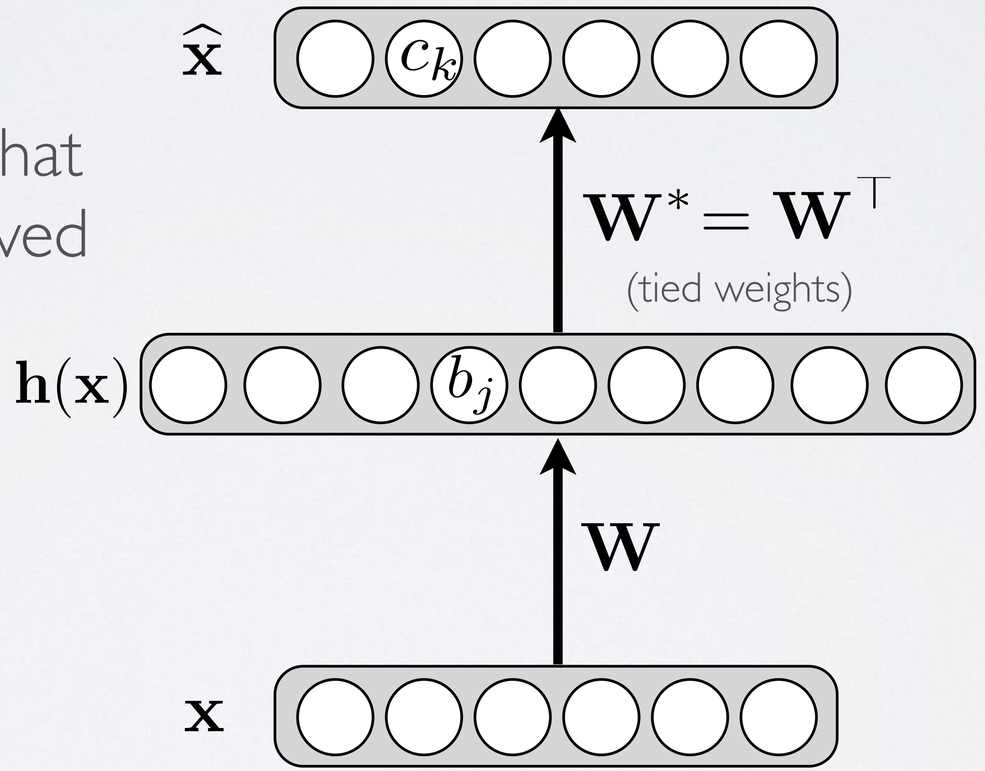
- 50% corrupted inputs



# CONTRACTIVE AUTOENCODER

**Topics:** contractive autoencoder

- Alternative approach to avoid uninteresting solutions
  - add an explicit term in the loss that penalizes that solution
- We wish to extract features that **only** reflect variations observed in the training set
  - we'd like to be invariant to the other variations



# CONTRACTIVE AUTOENCODER

**Topics:** contractive autoencoder

- New loss function:

$$\underbrace{l(f(\mathbf{x}^{(t)}))}_{\text{autoencoder reconstruction}} + \lambda \underbrace{||\nabla_{\mathbf{x}^{(t)}} \mathbf{h}(\mathbf{x}^{(t)})||_F^2}_{\text{Jacobian of encoder}}$$

  - where, for binary observations:

$$l(f(\mathbf{x}^{(t)})) = -\sum_k \left( x_k^{(t)} \log(\widehat{x}_k^{(t)}) + (1 - x_k^{(t)}) \log(1 - \widehat{x}_k^{(t)}) \right)$$

encoder keeps good information

$$||\nabla_{\mathbf{x}^{(t)}} \mathbf{h}(\mathbf{x}^{(t)})||_F^2 = \sum_j \sum_k \left( \frac{\partial h(\mathbf{x}^{(t)})_j}{\partial x_k^{(t)}} \right)^2$$

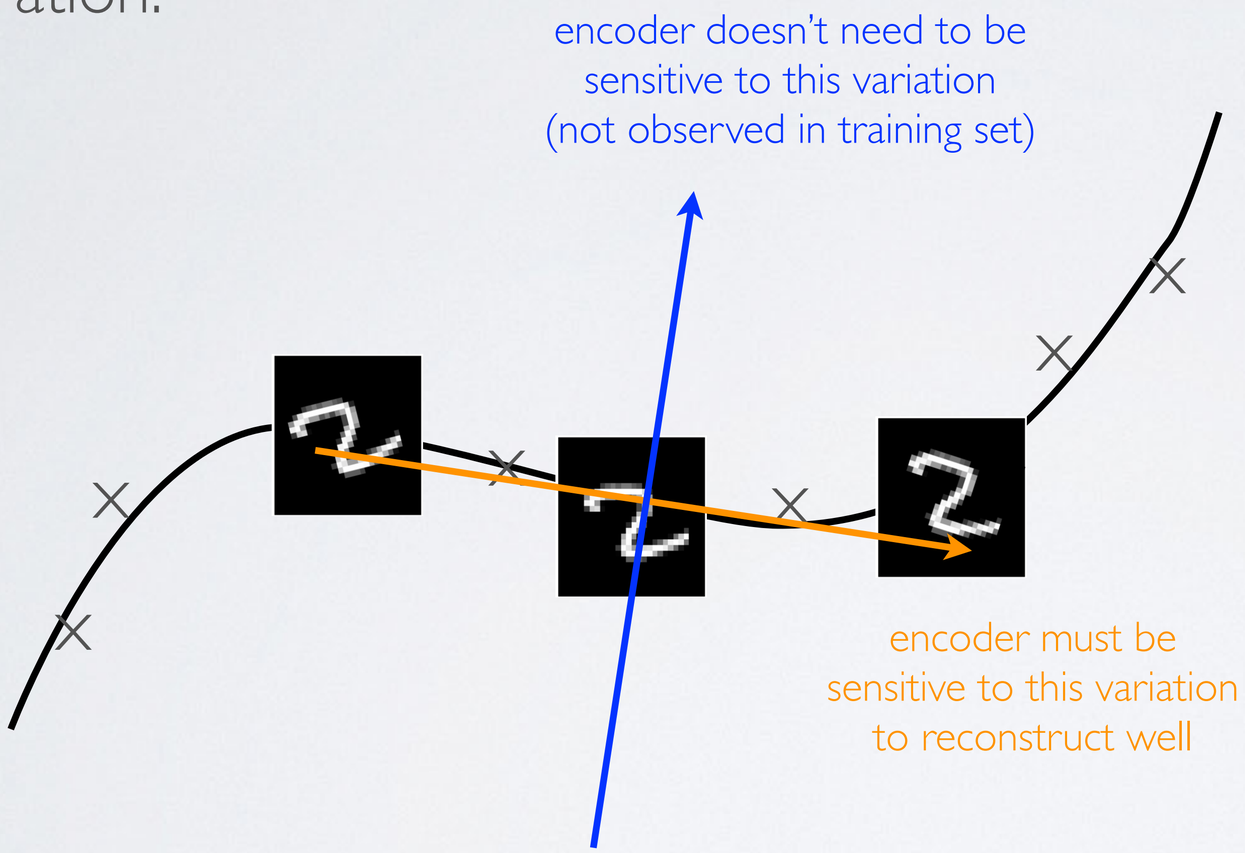
encoder throws away all information

encoder keeps **only** good information

# CONTRACTIVE AUTOENCODER

**Topics:** contractive autoencoder

- Illustration:

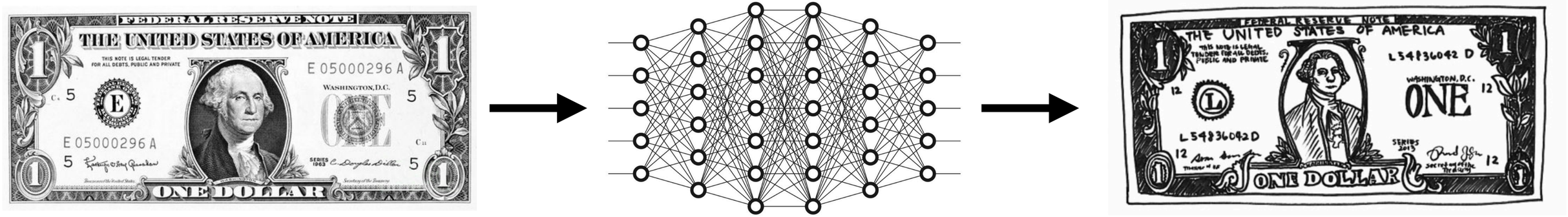


# WHICH AUTOENCODER ?

**Topics:** denoising autoencoder, contractive autoencoder

- Both the denoising and contractive autoencoder perform well
  - Advantage of denoising autoencoder: simpler to implement
    - requires adding one or two lines of code to regular autoencoder
    - no need to compute Jacobian of hidden layer
  - Advantage of contractive autoencoder: gradient is deterministic
    - can use second order optimizers (conjugate gradient, LBFGS, etc.)
    - might be more stable than denoising autoencoder, which uses a sampled gradient
- To learn more on contractive autoencoders:
  - Contractive Auto-Encoders: Explicit Invariance During Feature Extraction.
    Salah Rifai, Pascal Vincent, Xavier Muller, Xavier Glorot et Yoshua Bengio, 2011.

# Reconstruction error rarely leads to semantically useful representation.

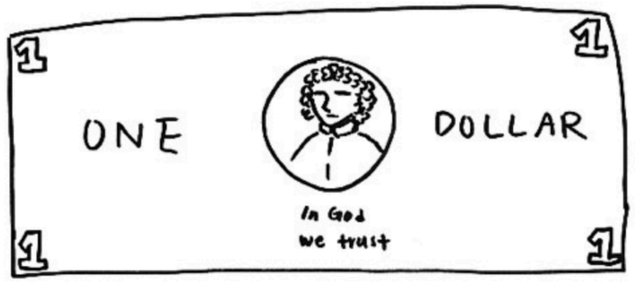



Reconstruction error as an autoencoder loss function:

$$\mathcal{L}(\ \text{  }\ ,\ \text{  }\ ) = \frac{1}{2}(\ \text{  }\ - \ \text{  }\ )^2$$

Dollar image source: Epstein, 2016; NN Image credit: Sambit Mahapatra

# Perspectives on self-supervised learning




*Drawing of a dollar bill from memory*


*Drawing subsequently made with a dollar bill present.*

- Drawing made from memory is different from the drawing made from an exemplar.
- Lesson: we don't retain a full representation of it. Or at least we can reproduce a detailed representation.
- Hypothesis: we really only retain enough features of the bill to distinguish it from other objects?
- Can we build representation learning algorithms that don't concentrate on pixel-level details, and only encode high-level features sufficient enough to distinguish different objects?

Image source: Epstein, 2016 via Ankesh Anand: https://ankeshanand.com/blog/2020/01/26/contrative-self-supervised-learning.html

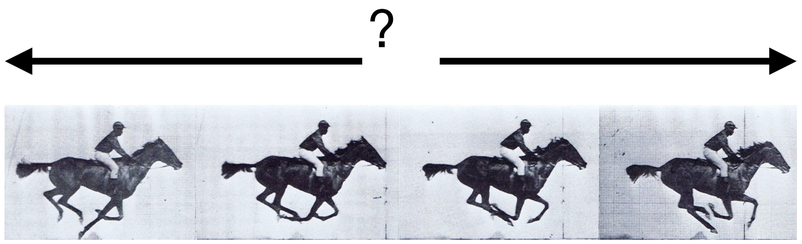
# What / why self-supervision?


**Self-supervision**: Recover useful/semantic representations by training models to answer specific questions about the data.

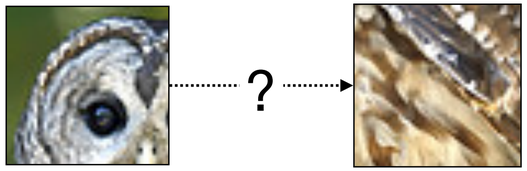
- **Good:**
  - Can procedurally generate potentially infinite amounts of annotation.
  - We can borrow tricks from supervised learning without labels.
  - Focus on only the information that you need (e.g., not pixels).
  - **Answering these questions requires more fundamental understanding of data.**
- **Not so good:** designing good questions also requires some fundamental understanding of the data (e.g., structure).

# Self-supervision in the wild

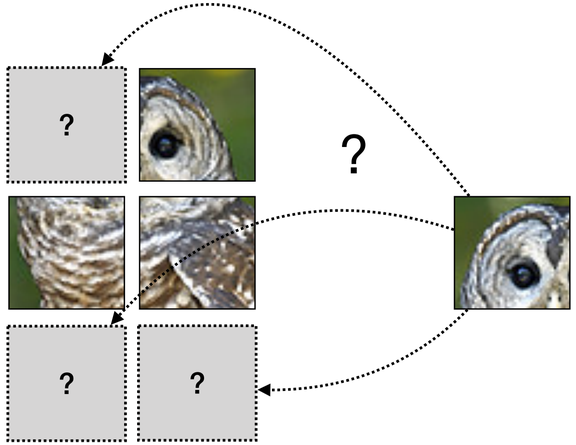
- **Question:** What direction is the video running?



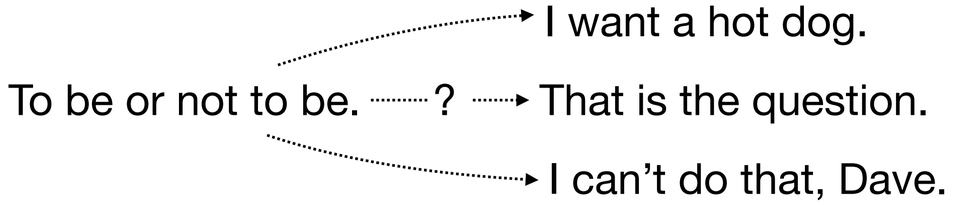

- **Question:** Do these image patches go together (context prediction)?




- **Question:** Where does this patch go (jigsaws)



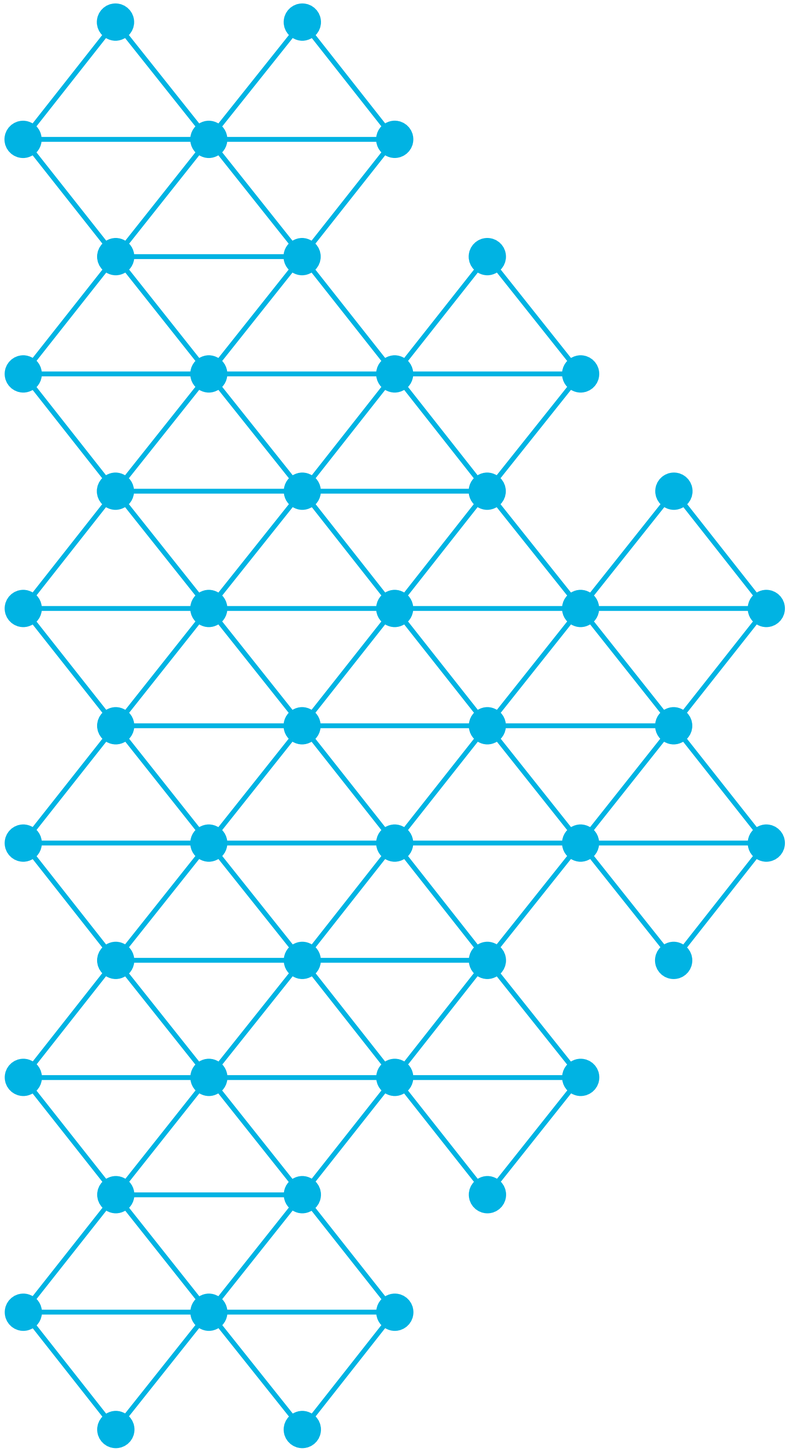

- **Question:** Which sentence follows this first one (Quick-thoughts)?



# Self-supervision in the wild


- Predict relative location of patches, solve jigsaw puzzles, predict other channels or modalities, inpaint, etc
- Two recent well-performing methods: predict an image transformation; predict patch-encoding


- Doersch, Gupta, and Efros. Unsupervised visual representation learning by context prediction.In ICCV 2015
- Pathak, Krähenbühl, Donahue, Darrell, and Efros. Context encoders: Feature learning by inpainting. In CVPR 2016
- Noroozi and Favaro. Unsupervised learning of visual representations by solving jigsaw puzzles. In ECCV 2016
- Zhang, Isola, and Efros. Split-brain autoencoders: Unsupervised learning by cross-channel prediction. In CVPR 2017



# Self-supervised learning as an instance of Transfer Learning

# Supervised Learning


- The **domain** $\mathcal{D}$ consists of: a *feature space* $\mathcal{X}$ and a *marginal probability distribution* $P(X)$, where $X = \{\boldsymbol{x}_1, ..., \boldsymbol{x}_n\} \in \mathcal{X}$.
- Given a specific domain, $\mathcal{D} = \{\mathcal{X}, P(X)\}$, the **supervised learning task** consists of two components: a label space $\mathcal{Y}$ and an objective predictive function $f(\cdot)$, denoted by $\mathcal{T} = \{\mathcal{Y}, f(\cdot)\}$.
- The task $\mathcal{T}$ defines a learning problem (the objective function is implicit) where **training data**, consisting of pairs $\{\boldsymbol{x}_i, y_i\}$, where $\boldsymbol{x}_i \in X$ and $y_i \in \mathcal{Y}$, is used to train the function $f(\cdot)$ to predict the corresponding labels of training instances.
- The function $f(\cdot)$ can be used to predict the corresponding label, $f(\boldsymbol{x})$, of a new instance $\boldsymbol{x}$.

**Supervised learning** aims to learning the predictive function $f(\cdot)$ in $\mathcal{D}$ using knowledge from the training data.

# Unsupervised Learning


- The **domain** $\mathcal{D}$ consists of: a *feature space* $\mathcal{X}$ and a *marginal probability distribution* $P(X)$, where $X = \{\boldsymbol{x}_1, ..., \boldsymbol{x}_n\} \in \mathcal{X}$.
- Given a specific domain, $\mathcal{D} = \{\mathcal{X}, P(X)\}$, the **unsupervised learning task** consists of simply an estimator function $f(\cdot)$ (denoted by $\mathcal{T} = \{f(\cdot)\}$), which is learned from the training data $\boldsymbol{x}_i \in X$.
- The function $f(\cdot)$ can be used to estimate statistics of $P(X)$ (including potentially estimating $P(X)$ itself).

**Unsupervised learning** aims to learn the estimator function $f(\cdot)$ in $\mathcal{D}$ using knowledge from the training data.

# Transfer Learning

- The **domain** $\mathcal{D}$ consists of: a *feature space* $\mathcal{X}$ and a *marginal probability distribution* $P(X)$, where $X = \{\boldsymbol{x}_1, ..., \boldsymbol{x}_n\} \in \mathcal{X}$.
- Given a specific domain, $\mathcal{D} = \{\mathcal{X}, P(X)\}$, a **task** consists of two components: a label space $\mathcal{Y}$ and an objective predictive function $f(\cdot)$ (denoted by $\mathcal{T} = \{\mathcal{Y}, f(\cdot)\}$), which is learned from the training data, consisting of pairs $\{\boldsymbol{x}_i, y_i\}$, where $\boldsymbol{x}_i \in X$ and $y_i \in \mathcal{Y}$.
- The function $f(\cdot)$ can be used to predict the corresponding label, $f(\boldsymbol{x})$, of a new instance $\boldsymbol{x}$.
- Given a **source domain** $\mathcal{D}_S$ and **learning task** $\mathcal{T}_S$, a **target domain** $\mathcal{D}_T$ and **learning task** $\mathcal{T}_T$:

> **Transfer learning** aims to help improve the learning of the target predictive function $f_T(\cdot)$ in $\mathcal{D}_T$ using the knowledge in $\mathcal{D}_S$ and $\mathcal{T}_S$, where $\mathcal{D}_S \neq \mathcal{D}_T$, or $\mathcal{T}_S \neq \mathcal{T}_T$ or both.

Lin, Yuan-Pin; Jung, Tzyy-Ping (27 June 2017). "Improving EEG-Based Emotion Classification Using Conditional Transfer Learning". *Frontiers in Human Neuroscience*. **11**: 334. From Wikipedia.

# Flavours of Transfer Learning

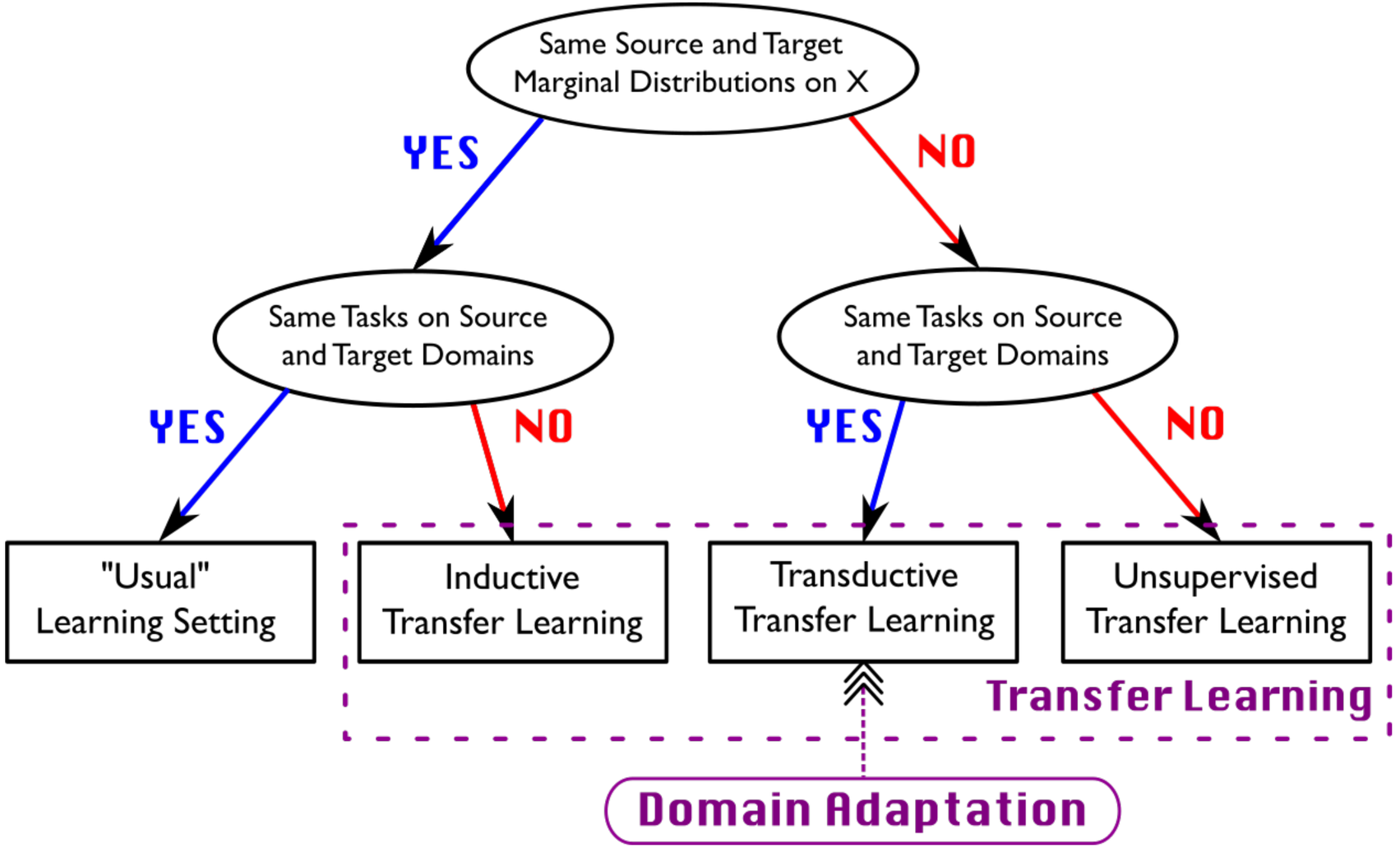



Image credit: Emilie Morvant - Own work, CC BY-SA 4.0, https://commons.wikimedia.org/w/index.php?curid=58812416

# Flavours of Transfer Learning

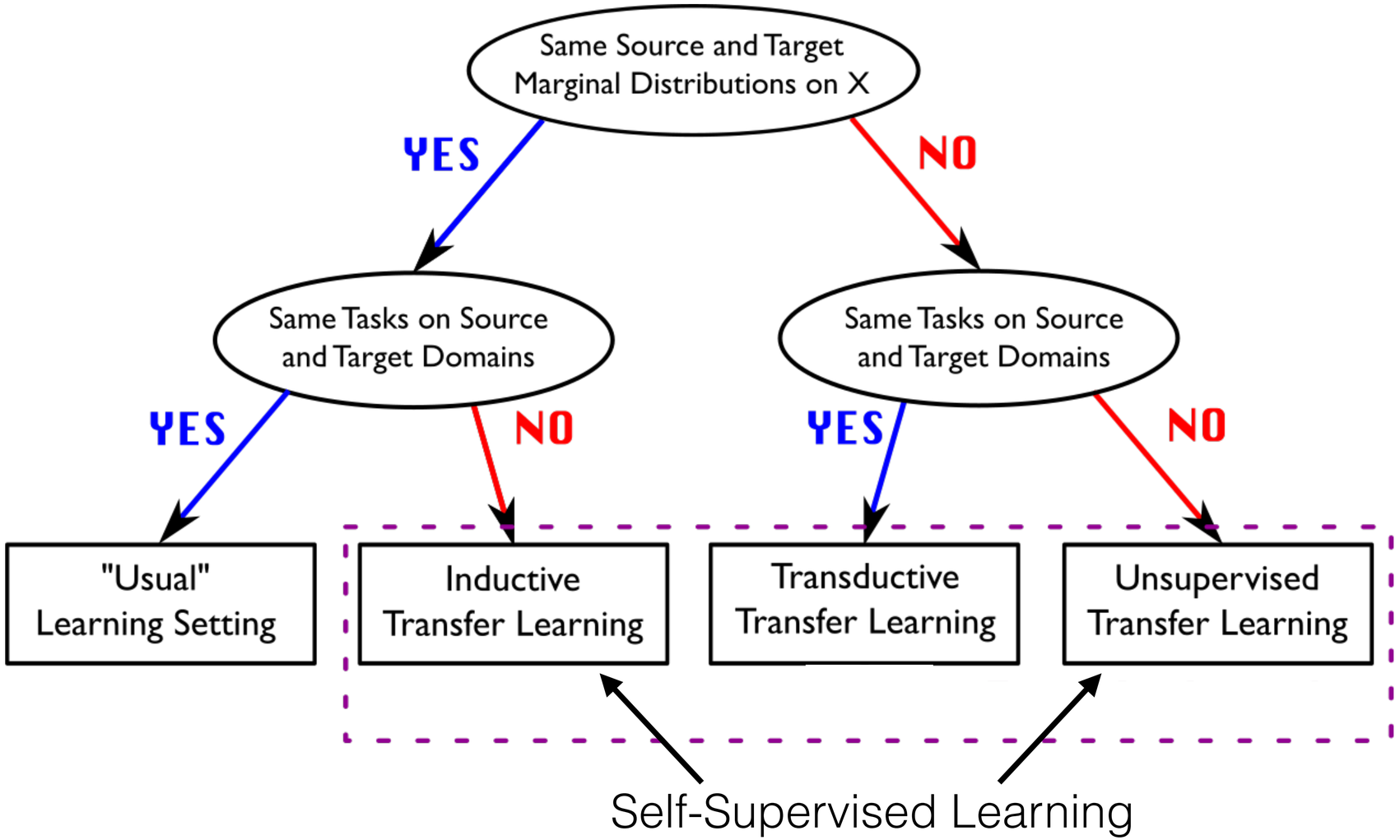



Image credit: Emilie Morvant - Own work, CC BY-SA 4.0, https://commons.wikimedia.org/w/index.php?curid=58812416

# A view on self-supervised learning


- Self-supervised learning is an instance of unsupervised learning methods in the sense that no external labels are needed to train the model parameters.

- Self-supervised learning is an instance of transfer learning where:

$$\mathcal{T}_S \neq \mathcal{T}_T$$

- Key characteristic of self-supervised learning is found in the definition of the source (training) task: $\mathcal{T}_S$

# A view on self-supervised learning


- For self-supervised learning, the source task, $\mathcal{T}_S$, is:

1. Unsupervised, I.e. it does not require external labels for training.

2. Designed to learn a representation that will extract discriminative (semantic) information from the input.

3. Designed to learn a representation that will possess appropriate invariances / equivariances
   - Eg. Invariance to low-level transformations of the input.

# IFT6268: self-supervised representation learning


- In this class, take a broad-view of self-supervised learning.
- We are interested in the wide-array of methods that exist to learn, without labels, useful / effective representations for downstream tasks (eg. RL tasks, classification, structure output prediction, etc.)