

Frontiers in deep learning

Mar 13, 2026

[Tailin Wu](#), Westlake University

Website:

<https://ai4s.lab.westlake.edu.cn/course/frontiers-2026/>



Image from: sdecoret

Outline

- Deep learning: fundamentals
 - Two foundational principles
 - Their realization in neural architecture and learning
- Optimization (SGD) and federative learning
 - Optimization with SGD
 - Federative learning

Previous class: a bird-eye view of deep learning

Tasks

- Classification/regression
- Simulation
- Inverse design/inverse problem
- Control/planning

×

Neural architecture

- Multilayer perceptron
- Graph Neural Networks
- Convolutional Neural Networks
- Transformers

×

Learning paradigm

- Supervised learning
- Generative modeling
- Foundation models
- Reinforcement learning
- Evolutionary and multi-objective optimization

Application (AI & Science)

- Robotics
- Games (e.g., Go, atari)
- Autonomous Driving
- PDEs
- Life science
- Materials science

Foundational principles in deep learning

What are the most **important insights** from 30 years of deep learning?

Foundational principles in deep learning 1

What are the most **important insights** from 30 years of deep learning?



Method 1: go from A to B in a **straight line**
(learn a **direct mapping** from A to B)

Hard to learn due to the complexity
of A, B and their difference!

Foundational principles in deep learning 1

What is the most **important insight** from 30 years of deep learning?



Method 2: go from A to B in **small, easier steps**
(**compose step-by-step** simple mappings to map A to B)

Much easier!

Foundational principles in deep learning 1

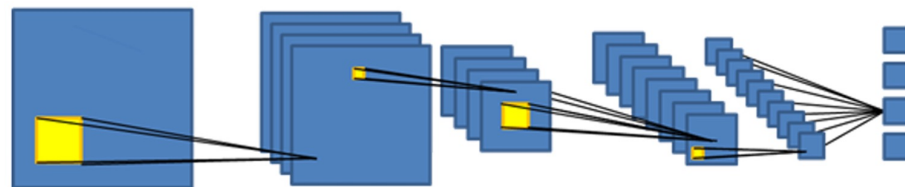
Insight: to construct a complex mapping from A to B, it is much easier to compose simple mappings



input

deep neural networks

target



“cat”

Foundational principles in deep learning 1

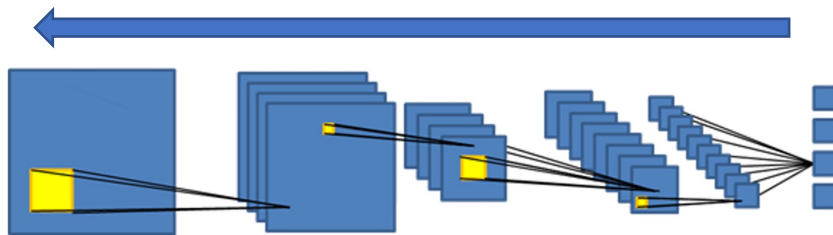
Insight: to construct a complex mapping from A to B, it is much easier to compose simple mappings



gradient w.r.t.
parameter

$$\frac{\partial L}{\partial \theta}$$

Backpropagation 反向传播



loss

$$L = \text{MSE}(f_{\theta}(x), y)$$

Foundational principles in deep learning 1

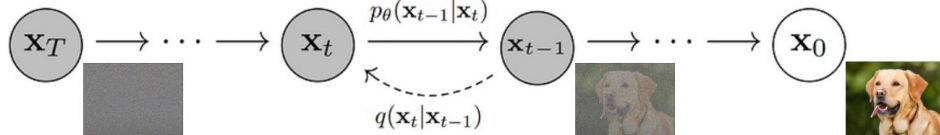
Insight: to construct a complex mapping from A to B, it is much easier to compose simple mappings



Gaussian distribution

diffusion model

data distribution

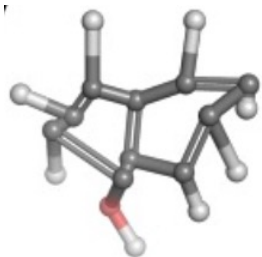


Foundational principles in deep learning 1

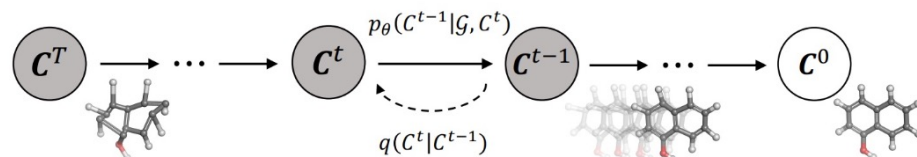
Insight: to construct a complex mapping from A to B, it is much easier to compose simple mappings



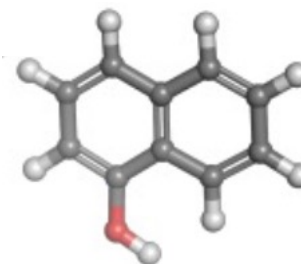
Gaussian distribution



diffusion model

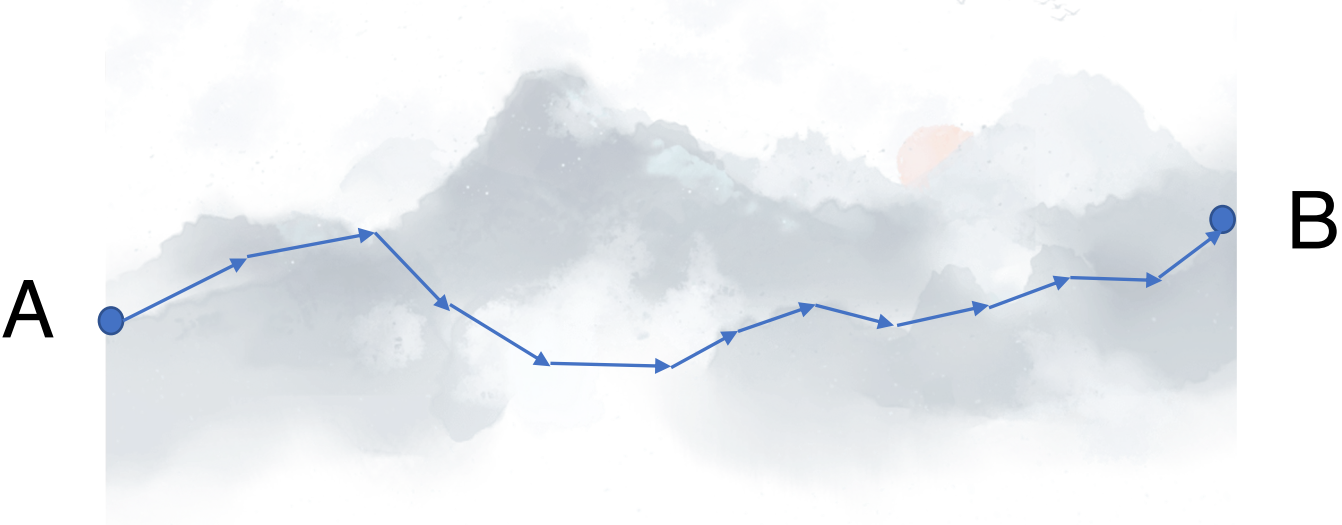


data distribution



Foundational principles in deep learning 1

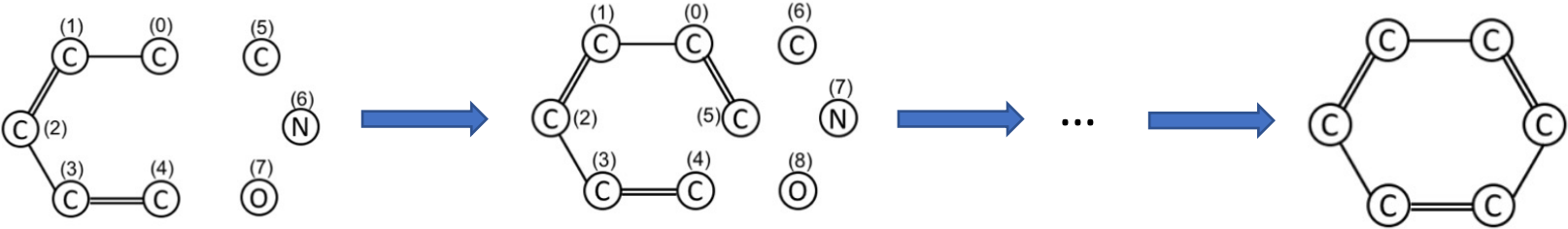
Insight: to construct a complex mapping from A to B, it is much easier to compose simple mappings



starting configuration

reinforcement learning

target configuration



Foundational principles in deep learning

What are the most **important insights** from 30 years of deep learning?

1. Model a hard transformation by composing many simple, easy transformations.

*This principle underlies all **neural architectures** and **learning paradigms***

2. Directly optimizing the final objective using probability and information theory

*Almost all **learning objectives** can be reduced to maximum likelihood or minimizing/maximizing information*

How to tackle a task using deep learning:

1. Specify the task (including input, target), and define its **learning objective**;
2. Choose appropriate **neural architecture** and **learning process**;
3. Train, evaluate (and iterate)

Foundational principles in deep learning

What are the most **important insights** from 30 years of deep learning?

1. Model a hard transformation by composing many simple, easy transformations.

*This principle underlies all **neural architectures** and **learning paradigms***

- Multilayer Perceptron (MLP)
- Backpropagation

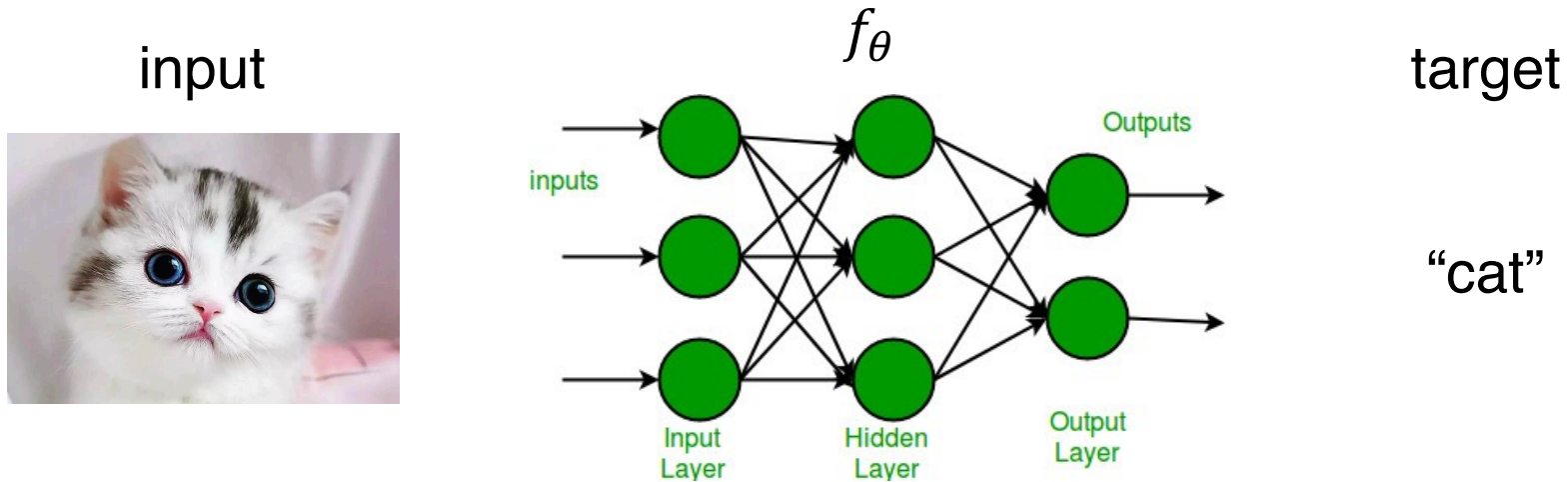
2. Directly optimizing the final objective using probability and information theory

Almost all learning objectives can be reduced to maximum likelihood or minimizing/maximizing information

Interactive notebook: [https://github.com/AI4Science-WestlakeU/frontiers in AI course](https://github.com/AI4Science-WestlakeU/frontiers_in_AI_course)



Multilayer Perceptron (MLP)



An MLP f_{θ} with 1 layer: $\sigma(W_1x + b_1)$: linear transformation with nonlinear activation

An MLP f_{θ} with n layers: $\sigma(W_n\sigma(\dots\sigma(W_2\sigma(W_1x + b_1) + b_2) \dots + b_n)$

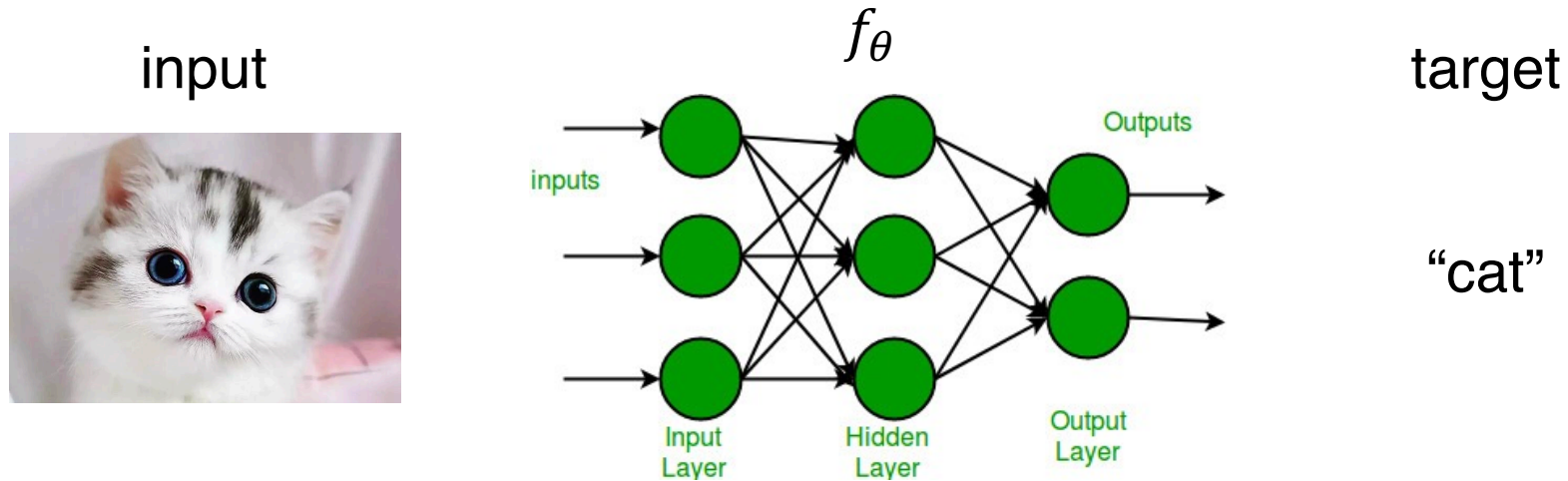
(Application of the **foundational principle 1**)

W_i : weight matrix to be learned

b_i : bias vector to be learned

σ : (nonlinear) activation function

MLP: universal approximation theorem



An MLP f_θ that has 1 hidden layer (with arbitrary width) and a nonlinear activation function can approximate any function to arbitrary precision [1][2].

Here $f_\theta(x) = W_2\sigma(W_1x + b_1)$

- With one hidden layer, may need **exponential** number of neurons w.r.t. input size
- With more layers, the neurons needed may be **polynomial** [3]

[1] Funahashi, Ken-Ichi. "On the approximate realization of continuous mappings by neural networks." Neural networks 2.3 (1989): 183-192.

[2] Hornik, Kurt, Maxwell Stinchcombe, and Halbert White. "Multilayer feedforward networks are universal approximators." Neural networks 2.5 (1989): 359-366.

[3] Rolnick, David, and Max Tegmark. "The power of deeper networks for expressing natural functions." ICLR 2018

Learning with gradient descent

$$f_{\theta}(x) = \sigma(W_n \sigma(\dots \sigma(W_2 \sigma(W_1 x + b_1) + b_2) \dots + b_n)$$

To fit dataset $\{(x_i, y_i)\}, i = 1, 2, \dots, N$, we can use Mean Squared Error (MSE):

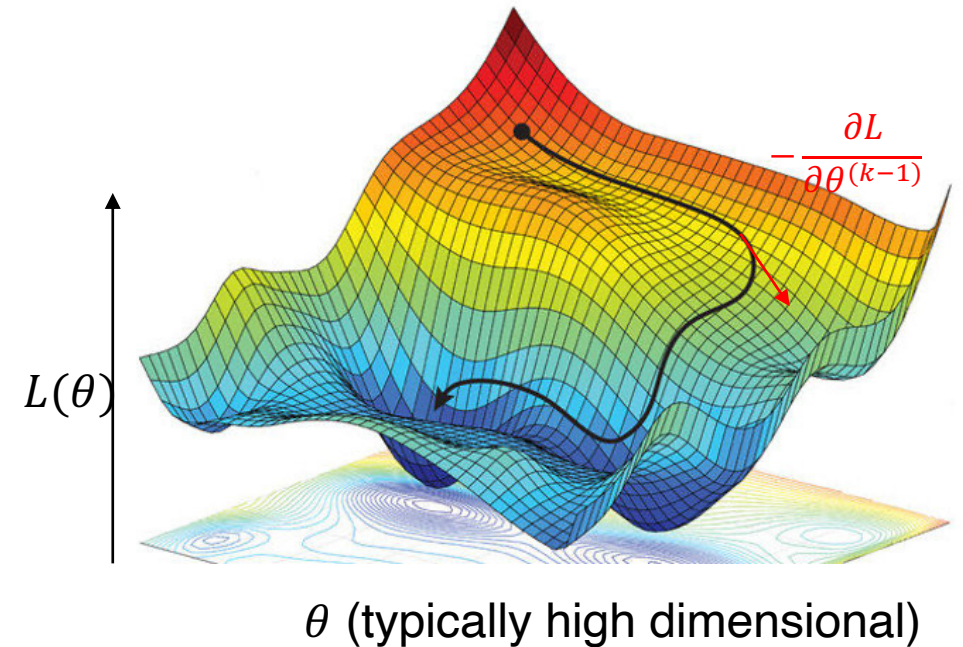
$$L(\theta) = \frac{1}{N} \sum_{i=1}^N (y_i - f_{\theta}(x_i))^2$$

How can we optimize the parameter $\theta = (W_1, \dots, W_n, b_1, \dots, b_n)$?

Answer: compute $\frac{\partial L}{\partial \theta}$, then we can perform gradient descent :

$$\theta^{(k)} \leftarrow \theta^{(k-1)} - \eta \frac{\partial L}{\partial \theta^{(k-1)}}$$

η : learning rate



Backpropagation

Let's take a two layer MLP $f_{\theta}(x) = W_2\sigma(W_1x + b_1)$ as an example:

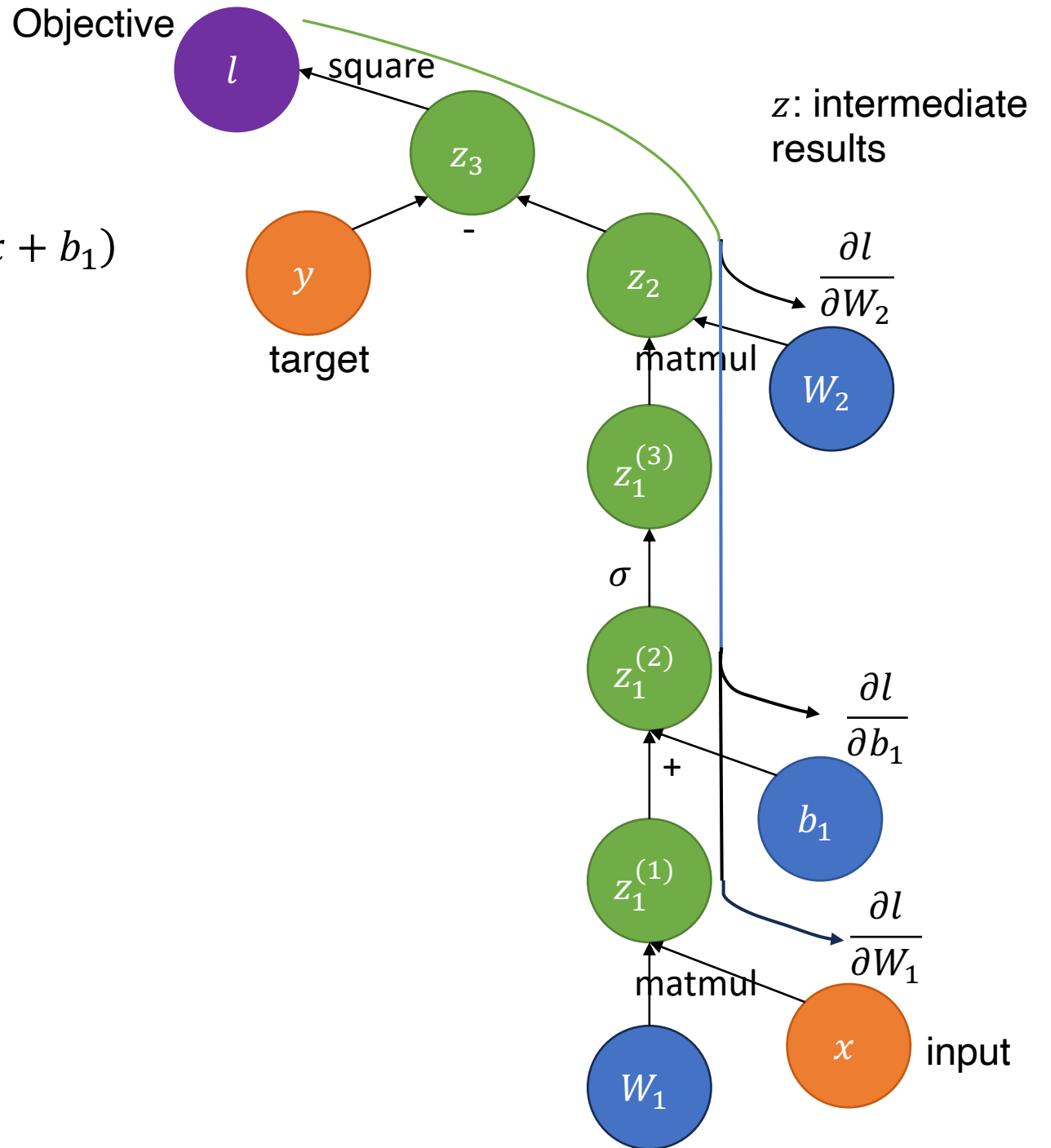
Objective: $l = (y - f_{\theta}(x))^2$

$$\frac{\partial l}{\partial W_2} = \frac{\partial l}{\partial z_3} \cdot \frac{\partial z_3}{\partial z_2} \cdot \frac{\partial z_2}{\partial W_2}$$

$$\frac{\partial l}{\partial b_1} = \frac{\partial l}{\partial z_3} \cdot \frac{\partial z_3}{\partial z_2} \cdot \frac{\partial z_2}{\partial z_1^{(3)}} \cdot \frac{\partial z_1^{(3)}}{\partial z_1^{(2)}} \cdot \frac{\partial z_1^{(2)}}{\partial b_1}$$

$$\frac{\partial l}{\partial W_1} = \frac{\partial l}{\partial z_3} \cdot \frac{\partial z_3}{\partial z_2} \cdot \frac{\partial z_2}{\partial z_1^{(3)}} \cdot \frac{\partial z_1^{(3)}}{\partial z_1^{(2)}} \cdot \frac{\partial z_1^{(2)}}{\partial z_1^{(1)}} \cdot \frac{\partial z_1^{(1)}}{\partial W_1}$$

shared, no need to recompute



Foundational principles in deep learning 1: summary

1. Model a hard transformation by composing many simple, easy transformations.

*This principle underlies all **neural architectures** and **learning paradigms***

- Multilayer Perceptron (MLP)
- Backpropagation
- Optimization with gradient descent

Foundational principles in deep learning

2. Directly optimizing the final objective using probability and information theory

Almost all learning objectives can be reduced to maximum likelihood or minimizing/maximizing information

Maximum likelihood objective underlies:

- **MSE loss**
- **Uncertainty quantification**
- Variational autoencoder (VAE)
- Diffusion model

Information-based objective underlies:

- **Cross-entropy loss**
- **Information Bottleneck**
- GAN, infoGAN
- Contrastive learning
- InfoMax: Deep Graph InfoMax
- Active learning
- Reinforcement learning:
 - Exploration vs. exploitation tradeoff, empowerment

Maximum likelihood

We have data $\{x_i\}, i = 1, \dots, N$, and want to use a probability model $p_\theta(x)$ to model it. Maximizing the likelihood is equivalent to minimizing the negative log-likelihood:

$$-\log P(\{x_i\}_{i=1}^N) = -\log \prod_{i=1}^N p_\theta(x_i) = -\sum_{i=1}^N \log p_\theta(x_i)$$

Maximum likelihood: deriving MSE

We have data $\{(x_i, y_i)\}, i = 1, \dots, N$, and want to use a probability model $p_\theta(y|x)$ to model it.

Here we assume $p_\theta(y|x) \sim \mathcal{N}(y; \mu_\theta(x), \sigma_\theta^2(x))$ is a conditional Gaussian:

$$p_\theta(y|x) = \frac{1}{\sqrt{2\pi}\sigma_\theta(x)} e^{-\frac{(y-\mu_\theta(x))^2}{2\sigma_\theta^2(x)}}$$

We have

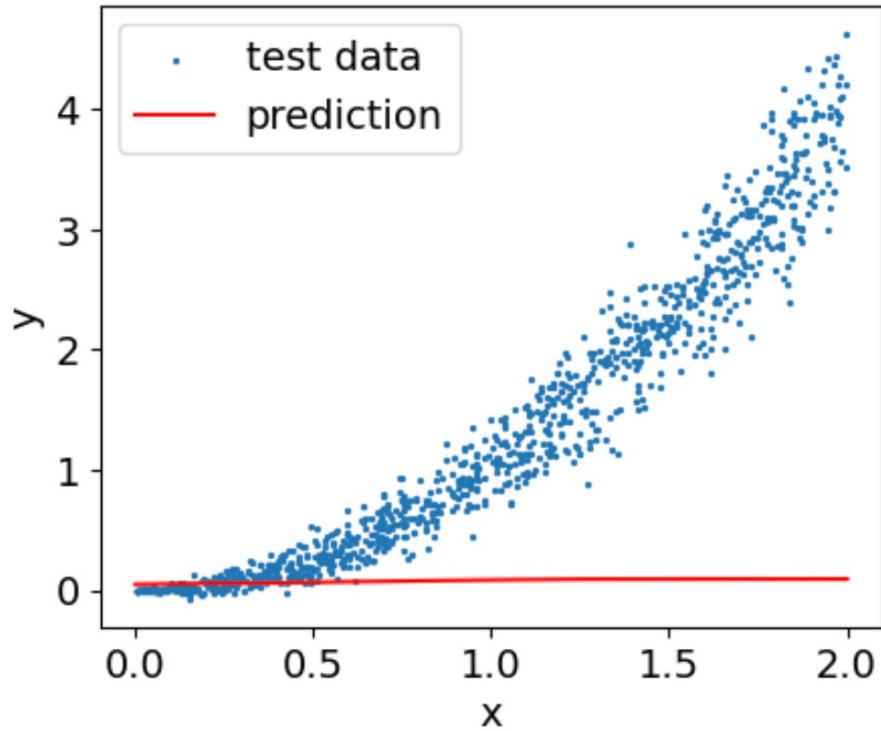
$$\begin{aligned} -\log P(Y|X) &= -\log \prod_{i=1}^N p_\theta(y_i|x_i) = -\sum_{i=1}^N \log p_\theta(y_i|x_i) \\ &= \sum_{i=1}^N \left[\frac{(y_i - \mu_\theta(x_i))^2}{2\sigma_\theta^2(x_i)} + \log \sigma_\theta(x_i) \right] \end{aligned}$$

$$X = \{x_i\}_{i=1}^N, Y = \{y_i\}_{i=1}^N$$

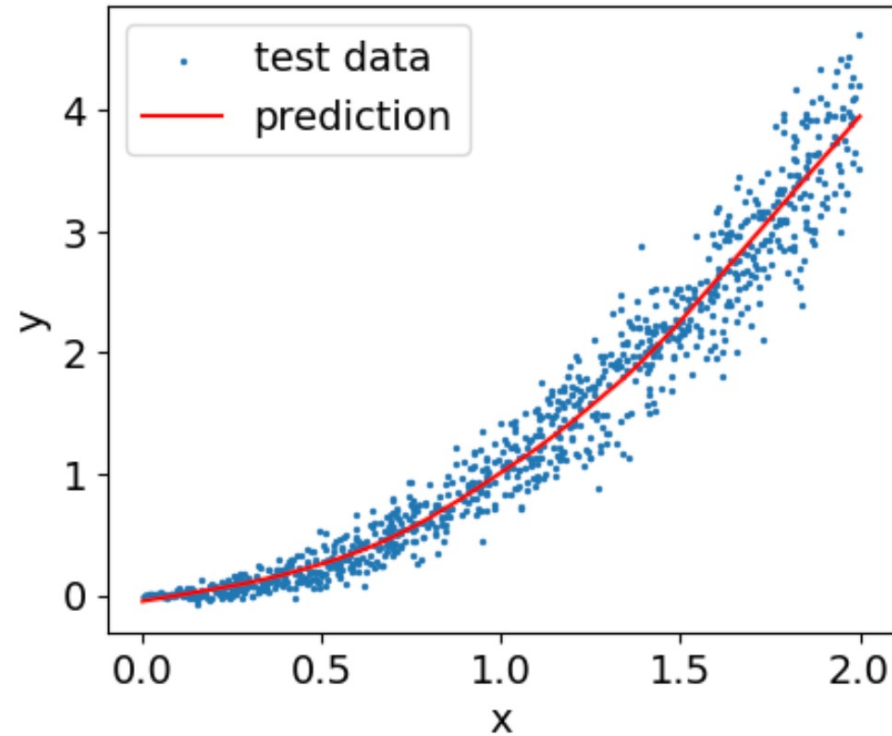
Assuming $\sigma_\theta(x) \equiv 1$, we have $-\log P(Y|X) = \frac{1}{2} \sum_{i=1}^N (y_i - \mu_\theta(x_i))^2$ **MSE loss**

Maximum likelihood: deriving MSE

Prediction by initial model f_θ :



Prediction after training f_θ :

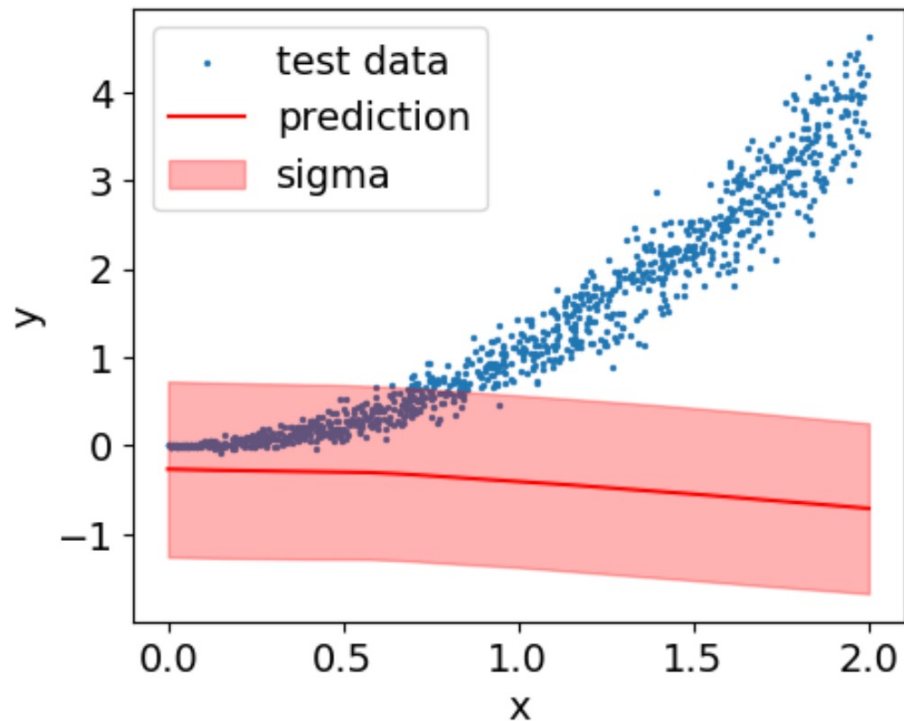


Maximum likelihood: estimating uncertainty

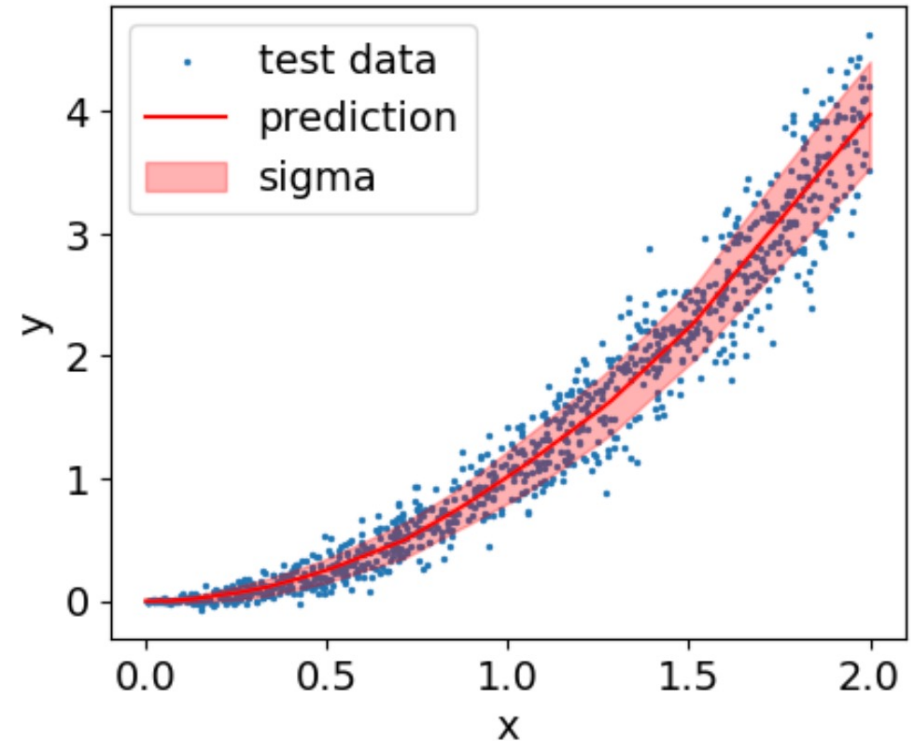
If $\sigma_\theta(x)$ can be learned, we can also estimate uncertainty [1]:

$$-\log P(Y|X) = \sum_{i=1}^N \left[\frac{(y - \mu_\theta(x))^2}{2\sigma_\theta^2(x)} + \log \sigma_\theta(x) \right]$$

Prediction by initial model f_θ :



Prediction by trained model f_θ :



Foundational principles in deep learning

2. Directly optimizing the final objective using probability and information theory

Almost all learning objectives can be reduced to maximum likelihood or minimizing/maximizing information

Maximum likelihood objective underlies:

- MSE loss
- Uncertainty quantification
- Variational autoencoder (VAE)
- Diffusion model

Information-based objective underlies:

- Cross-entropy loss
- **Information Bottleneck**
- GAN, infoGAN
- Contrastive learning
- InfoMax: Deep Graph InfoMax
- Active learning
- Reinforcement learning:
 - Exploration vs. exploitation tradeoff, empowerment

Information diagram

A type of Venn diagram to illustrate relationships among Shannon's basic measures of information for (multiple) variables.

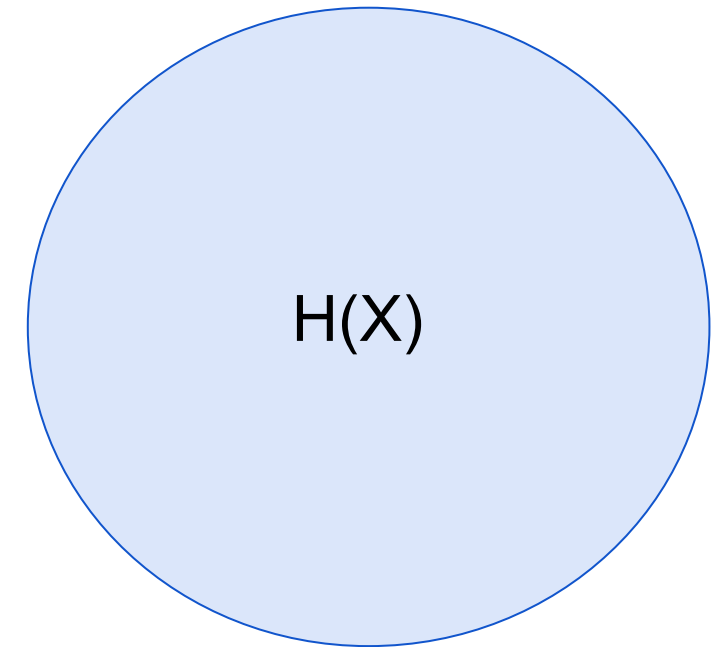
H(X): entropy of variable X, means the expected amount of information conveyed by identifying the outcome of a random sampling.

E.g. if X is a categorical variable taking values in $X \in \{1,2,3\}$ with probability of $\{1/4, 1/4, 1/2\}$.

When we draw a sample, and get e.g. $X=1$, the probability of it happening is $P(X=1)=1/4$, this event gains us $\log_2 \frac{1}{P(X=1)} = 2$ (bits) of information.

$$\text{Entropy: } H(X) = \sum_x P(X = x) \log_2 \frac{1}{P(X = x)}$$

Alternatively, we can understand it as the amount of information needed to deterministically specify a random variable.



Information diagram

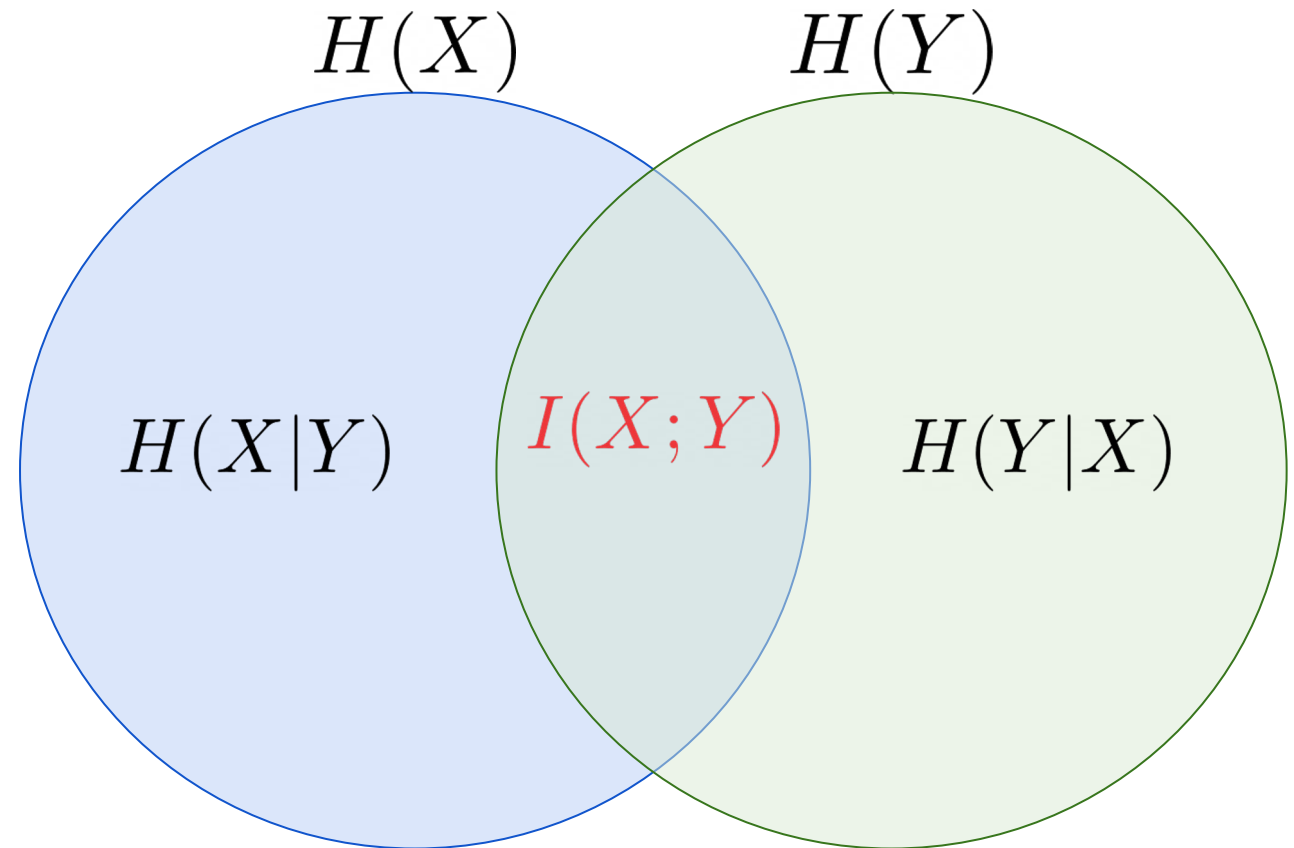
For multiple variables, we can treat each circle as a “set”.

$H(X|Y)$: entropy of X conditioned on Y

Meaning: given Y, how much more information needed to fully specify X.

$I(X; Y)$: mutual information between X and Y

Meaning: how much information obtained about X by observing Y



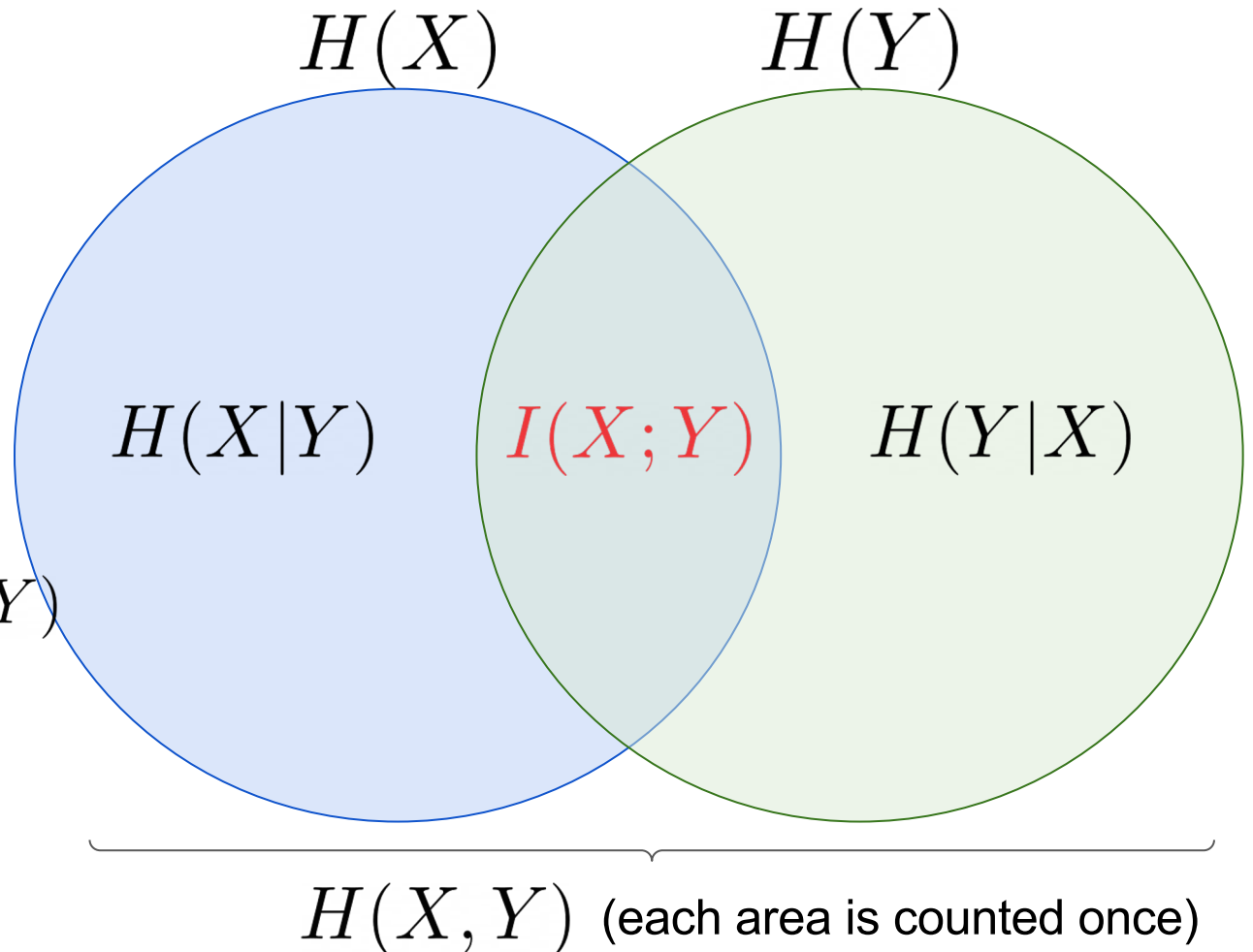
Information diagram

For multiple variables, we can treat each circle as a “set”.

Using set operations in the Venn diagram, we can easily derive:

$$H(X|Y) = H(X, Y) - H(Y)$$

$$\begin{aligned} I(X; Y) &= H(X) - H(X|Y) \\ &= H(Y) - H(Y|X) \\ &= H(X) + H(Y) - H(X, Y) \end{aligned}$$



Information diagram: quiz

If Y is a deterministic function of X , how does the information diagram look like?

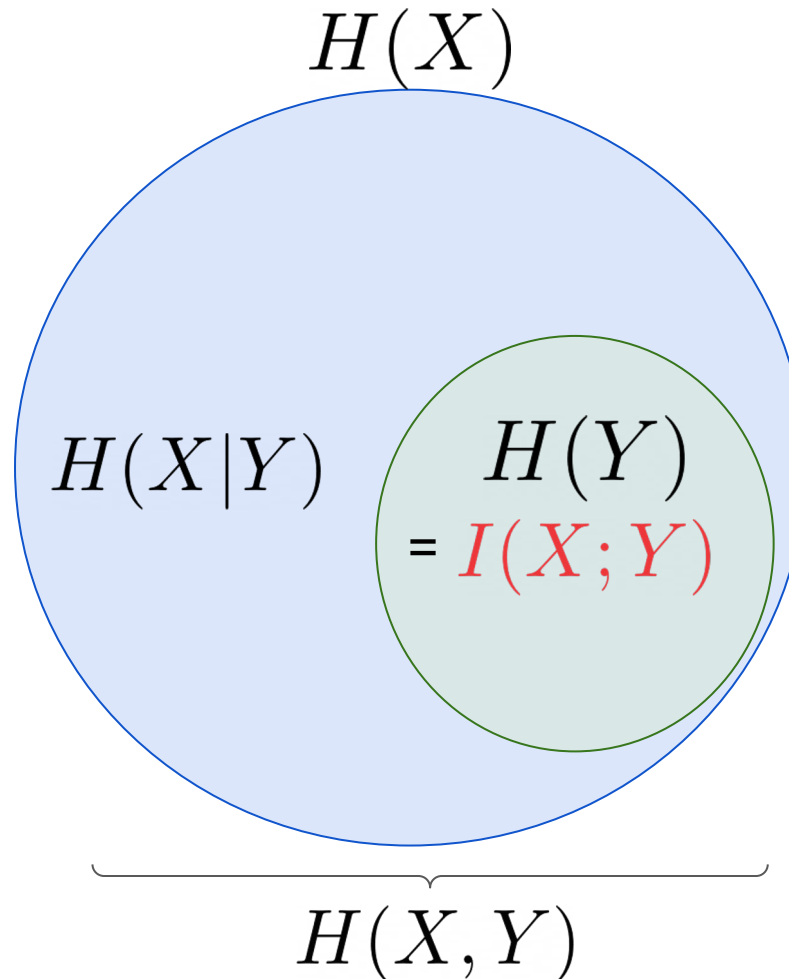
$$H(Y|X) = 0$$

Given X , the amount of information needed to specify Y is 0.

We can then easily derive:

$$H(X, Y) = H(X)$$

$$I(X; Y) = H(Y)$$



Information diagram: more variables

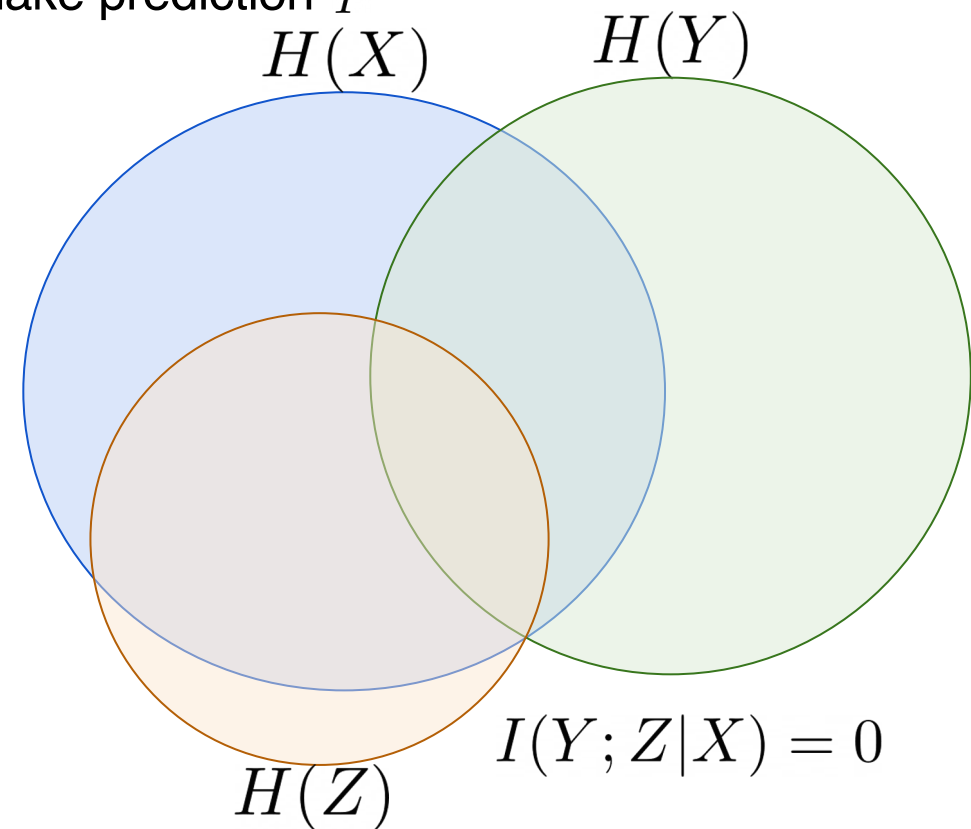
Procedure: (1) Specify dependence between variables; (2) Draw information diagram

Scenario: we have variables of X (images) and Y (labels). We also have a neural network that maps X to latent representation Z , based on which we make prediction \hat{Y}

Dependence: $\hat{Y} - Z - X - Y$

Since Z is a function of X , we have:
conditioned on X , Z is independent of Y :

$$I(Y; Z|X) = 0$$



How to optimize the information-based objective?

To maximize some quantity Q that is hard to optimize, we can maximize a learnable quantity \tilde{Q} that is less than Q (similar goes for minimizing)

Example: Evidence Lower Bound (ELBO) in variational autoencoder (VAE) [1], which is a lower bound for the log-likelihood of data.

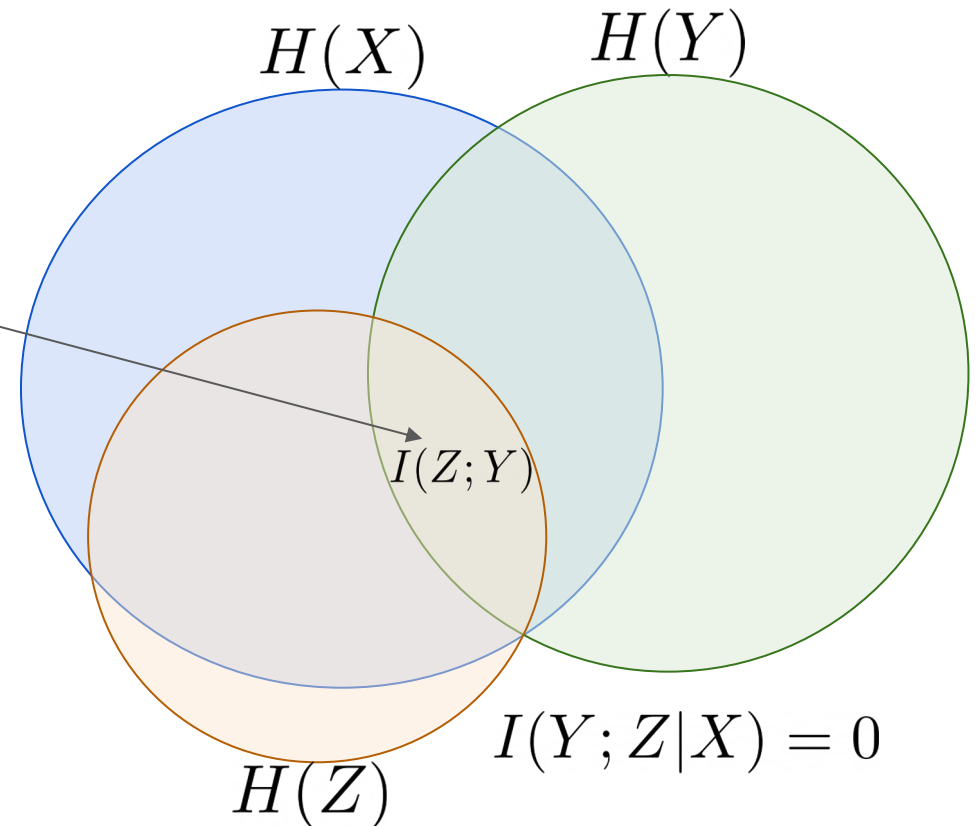
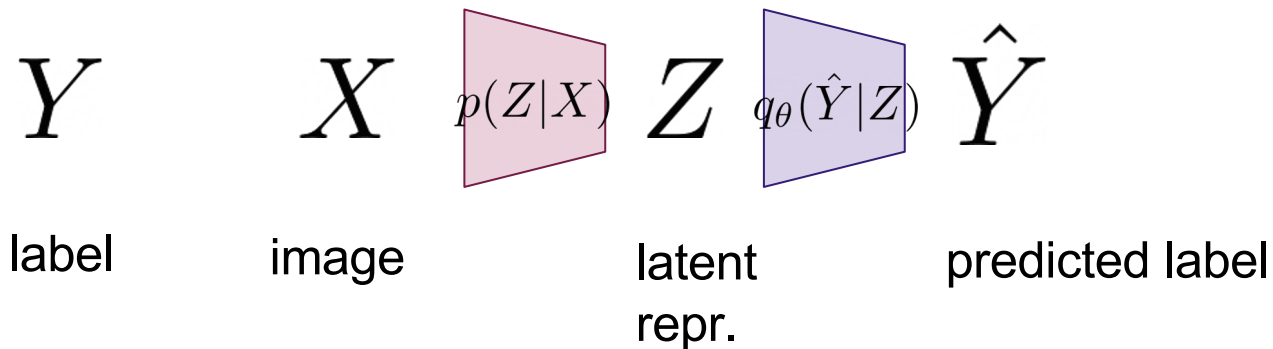
[1] Kingma, Diederik P., and Max Welling. "Auto-encoding variational bayes." arXiv preprint arXiv:1312.6114 (2013).

Maximizing mutual information

X are images, Y are corresponding labels. We have an encoder that takes as input X and outputs representation Z, which then predict label \hat{Y}

Dependence: $Y - X - Z - \hat{Y}$

Objective: $\max_Z I(Z; Y)$



Maximizing mutual information: cross-entropy

X are images, Y are corresponding labels. We have an encoder that takes as input X and outputs representation Z, which then predict label \hat{Y}

$$\begin{aligned} I(Y; Z) &:= \int dydz p(y, z) \log \frac{p(y|z)}{p(y)} \quad (\text{definition}) \\ &= \int dydz p(y, z) \log \frac{q_\theta(y|z)}{p(y)} + \underbrace{\int dydz p(y, z) \log \frac{p(y|z)}{q_\theta(y|z)}}_{= KL[p(y|z); q_\theta(y|z)] \geq 0} \\ &\geq \int dydz p(y, z) \log \frac{q_\theta(y|z)}{p(y)} \\ &= \int dydz p(y, z) \log q_\theta(y|z) + H(Y) \\ &= \int dydz \underbrace{dx p(x)p(y|x)p(z|x)}_{\text{(negative of cross-entropy!)}} \log q_\theta(y|z) + H(Y) \end{aligned}$$

Ignoring the constant $H(Y)$, we are maximizing $\mathbb{E}_{x,y \sim p(x,y), z \sim p(z|x)} [\log q_\theta(y|z)]$

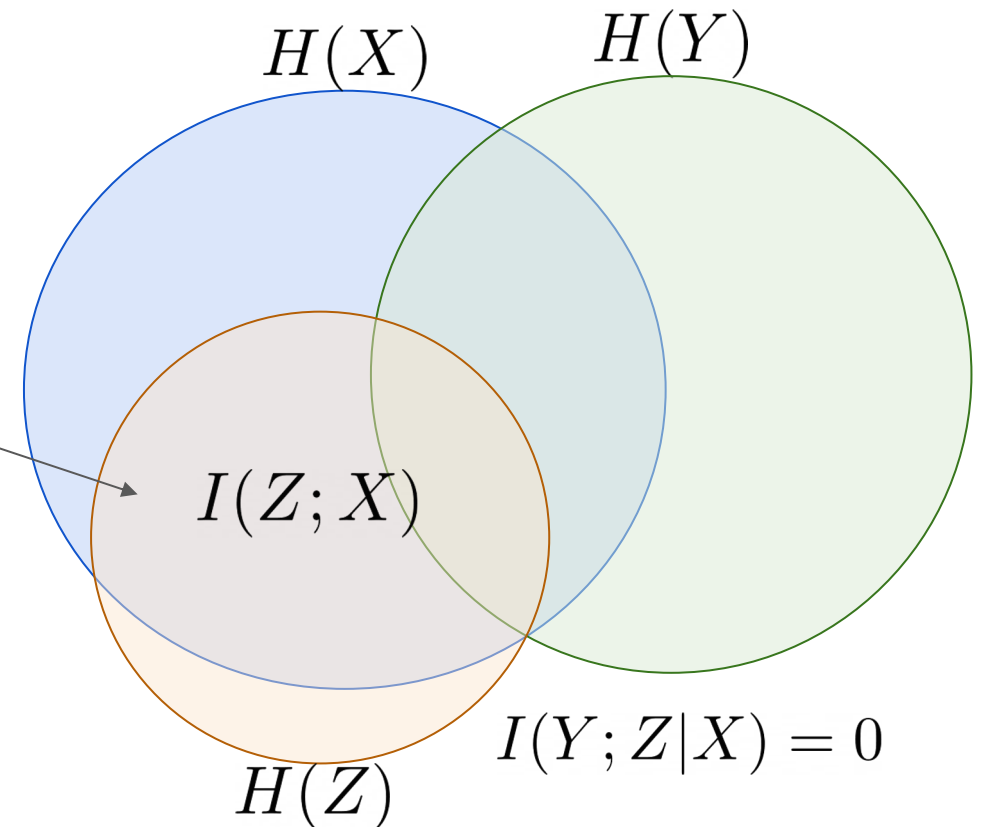
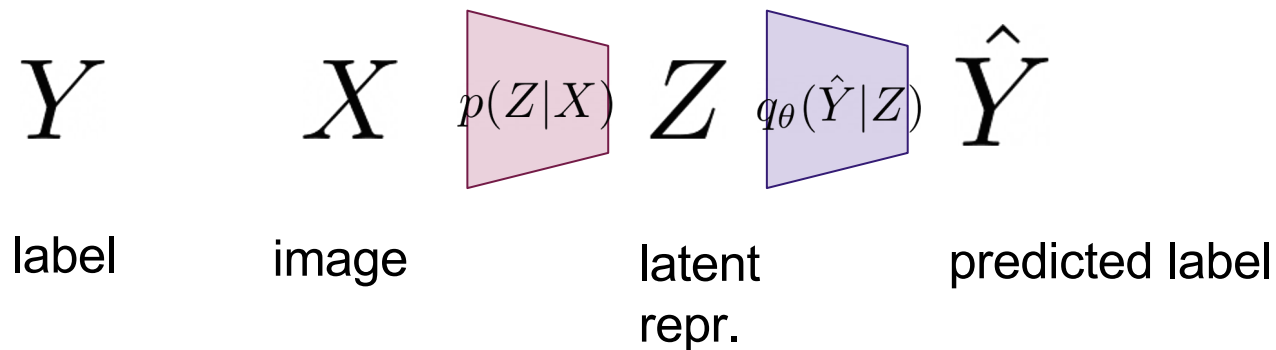
Minimizing mutual information

X are images, Y are corresponding labels. We have an encoder that takes as input X and outputs representation Z , which then predict label \hat{Y}

Dependence: $Y - X - Z - \hat{Y}$

Objective: $\min_Z I(Z; X)$

so that Z contains *less* information that is not relevant to predict Y : improving robustness and generalization



Minimizing mutual information

X are images, Y are corresponding labels. We have an encoder that takes as input X and outputs representation Z, which then predict label \hat{Y}

$$\begin{aligned} I(Z; X) &:= \int dx dz p(x, z) \log \frac{p(z|x)}{p(z)} \quad (\text{definition}) \\ &= \int dx dz p(x, z) \log \frac{p(z|x)}{r(z)} - \underbrace{\int dx dz p(x, z) \log \frac{p(z)}{r(z)}}_{= KL[p(z); r(z)] \geq 0} \\ &\leq \int dx dz p(x, z) \log \frac{p(z|x)}{r(z)} \\ &\simeq \frac{1}{N} \sum_{i=1}^N p(z|x_n) \log \frac{p(z|x_n)}{r(z)} \quad (\text{Monte Carlo estimation of the integral}) \end{aligned}$$

$r(z)$ can be approximated by a Gaussian or mixture of Gaussian, similar to the prior term in VAE

Information Bottleneck [1][2]

$$\min L = I(Z; X) - \beta \cdot I(Y; Z)$$

$$\approx \frac{1}{N} \sum_{i=1}^N \mathbb{E}_{z \sim p(z|x_i)} \left[-\beta \cdot p_{\theta}(z|x_i) \log \frac{p_{\theta}(z|x_i)}{r_{\theta}(z)} + \log q_{\theta}(y_i|z) \right]$$

Application of Information Bottleneck:

- Robust against adversarial attacks [2][3]
- Learning invariant and disentangled representations [4]
- RL:
 - Improving generalization [5]
 - Facilitating skill discovery [6]
 - Learning goal-conditioned policy [7]

[1] Tishby, Naftali, Fernando C. Pereira, and William Bialek. "The information bottleneck method." *arXiv preprint physics/0004057* (2000).

[2] Alemi, Alexander A., et al. "Deep variational information bottleneck." ICLR 2017.

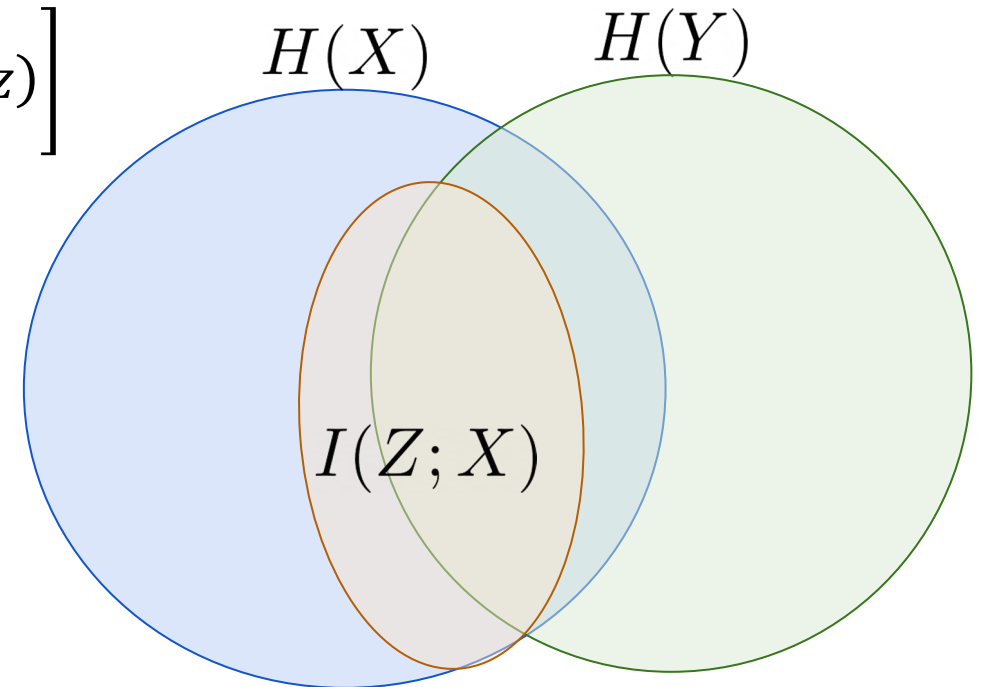
[3] Wu, Tailin, et al. "Graph information bottleneck." *NeurIPS 2020*.

[4] Achille, Alessandro, and Stefano Soatto. "Emergence of invariance and disentanglement in deep representations." *JMLR* 19.1 (2018): 1947-1980.

[5] Lu, Xingyu, et al. "Dynamics generalization via information bottleneck in deep reinforcement learning." *arXiv preprint arXiv:2008.00614* (2020).

[6] Sharma, Archit, et al. "Dynamics-aware unsupervised discovery of skills." *arXiv preprint arXiv:1907.01657* (2019).

[7] Goyal, Anirudh, et al. "Infobot: Transfer and exploration via the information bottleneck." *arXiv preprint arXiv:1901.10902* (2019).



Foundational principles in deep learning

2. Directly optimizing the final objective using probability and information theory

Almost all learning objectives can be reduced to maximum likelihood or minimizing/maximizing information

Maximum likelihood objective underlies:

- **MSE loss**
- **Uncertainty quantification**
- Variational autoencoder (VAE)
- Diffusion model

Information-based objective underlies:

- **Cross-entropy loss**
- **Information Bottleneck**
- GAN, infoGAN
- Contrastive learning
- InfoMax: Deep Graph InfoMax
- Active learning
- Reinforcement learning:
 - Exploration vs. exploitation tradeoff, empowerment

Foundational principles in deep learning: Summary

1. Model a hard transformation by composing many simple, easy transformations.
 - Multilayer Perceptron (MLP)
 - Backpropagation
2. Directly optimizing the final objective using probability and information theory
 - Maximum likelihood: MSE, uncertainty estimation
 - Information: cross-entropy, Information Bottleneck