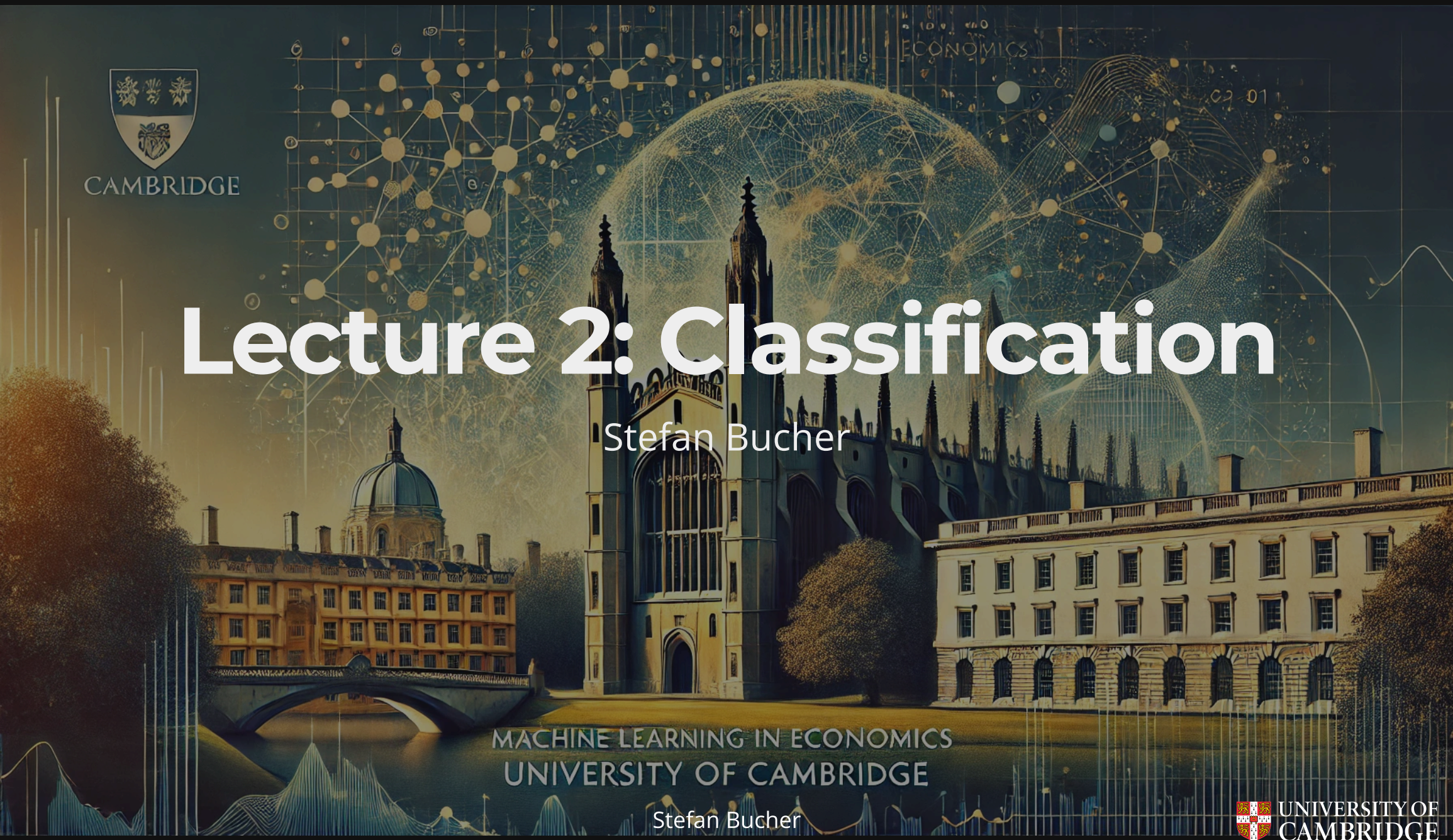


Lecture 2: Classification

Stefan Bucher

Open in Colab

Prince (2023, chaps. 5, 9).[1]

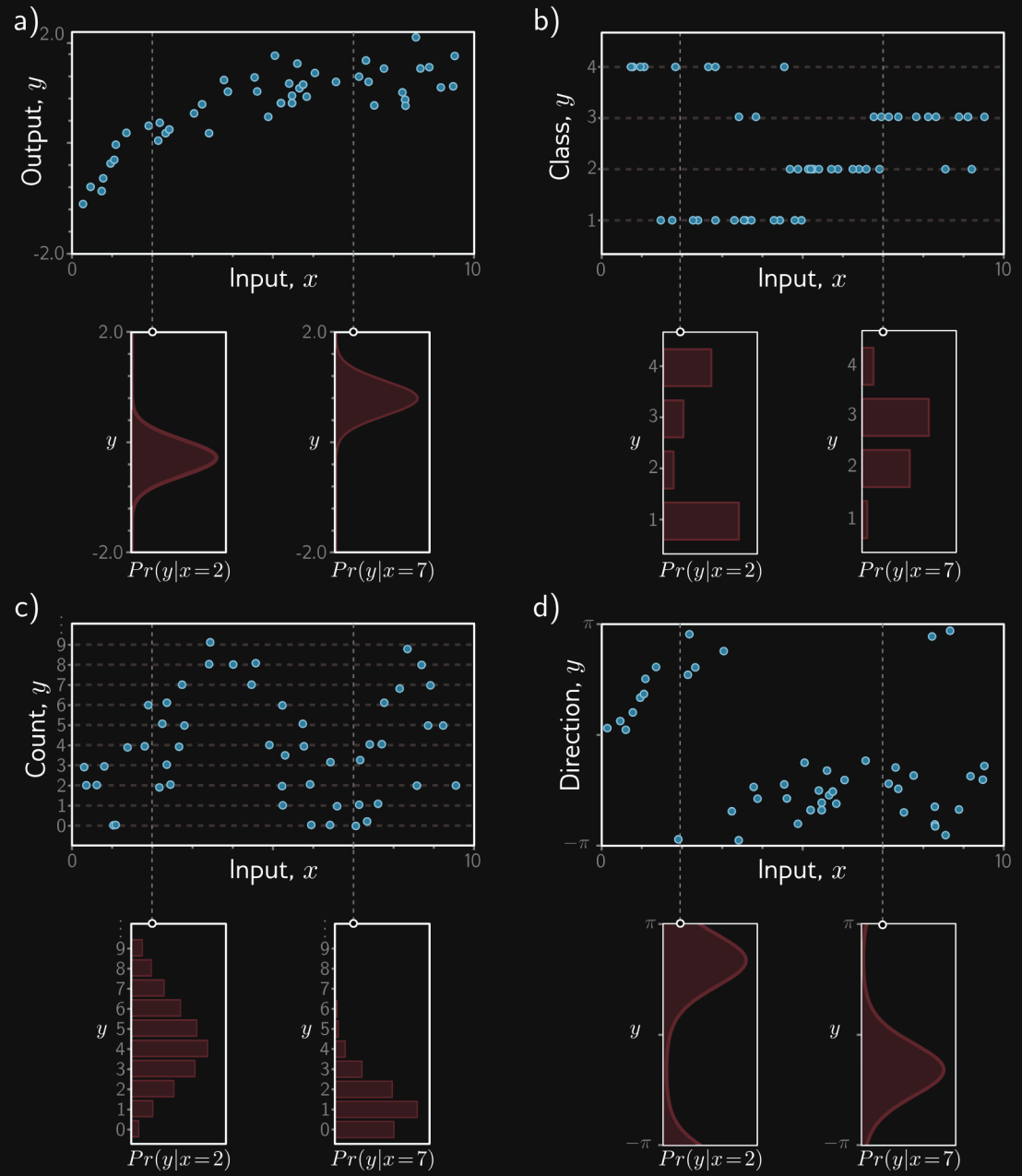
1. Figures taken or adapted from Prince (2023). All rights belong to the original author and publisher. These materials are intended solely for educational purposes.

Loss Functions

Prince (2023, chap. 5)

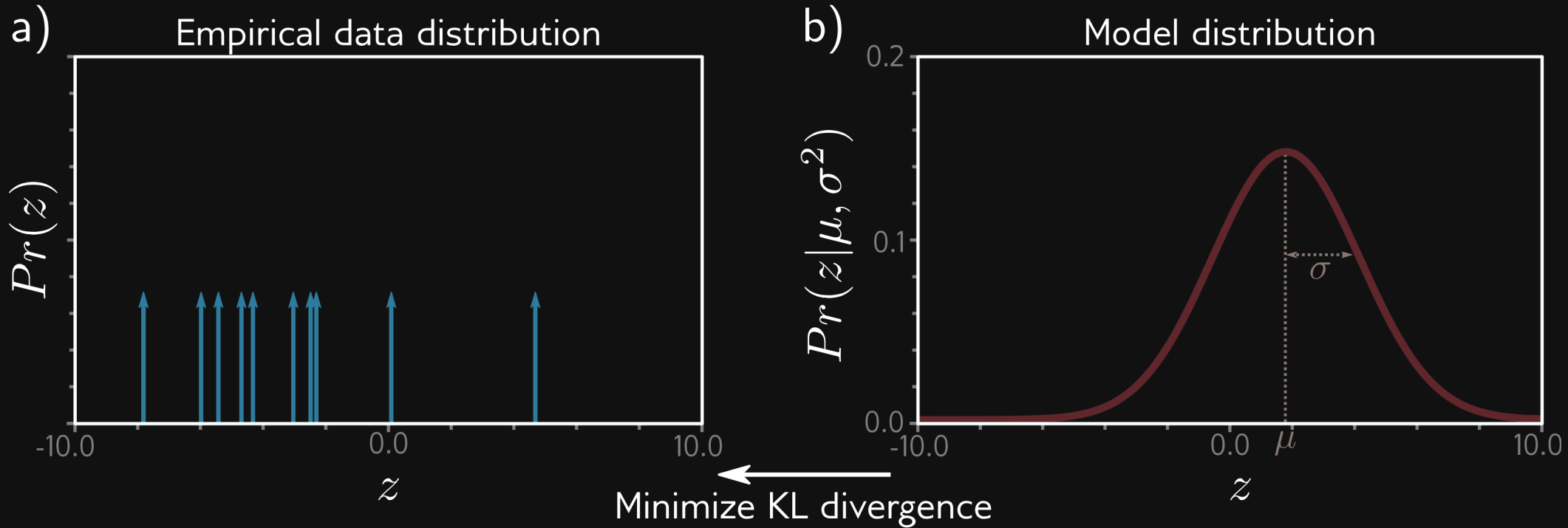
What is a Model?

> “A model is a probability distribution over a sequence of [observable] random variables.” — Thomas Sargent



What is a Good Model?

Model Q that is *close* to the true distribution P (of which we observe an empirical sample).



Cross-Entropy Loss (= Max LLH)

Minimizing KL div. from model $Q_\theta(y|x)$ to empirical $P(y|x)$

$$
\begin{aligned}
\hat{\theta} &= \arg\min_\theta D_{KL}[P||Q] \\
&= \arg\min_\theta \int_{-\infty}^{\infty} p(y|x) \log \frac{p(y|x)}{q_\theta(y|x)} dy \\
&= \arg\min_\theta - \int_{-\infty}^{\infty} p(y|x) \log q_\theta(y|x) dy \\
&= \arg\min_\theta - \int_{-\infty}^{\infty} \left(\frac{1}{I} \sum_{i=1}^{I} \delta[y - y_i] \right) \log q_\theta(y|x) dy \\
&= \arg\min_\theta - \sum_{i=1}^{I} \log q_\theta(y_i|x_i)
\end{aligned}
$$

is equivalent to minimizing the negative log-likelihood

$$\hat{\phi} = \arg\min_{\phi} -\sum_{i=1}^{I} \log q(y_i|\theta_i = f[x_i, \phi])$$

Distribution parametrized by ML model.

Maximizing Log-Likelihood

$$\hat{\phi} = \arg\max_{\phi} \prod_{i=1}^{I} q(y_i|\theta_i = f[x_i, \phi])$$

$$= \arg\max_{\phi} \log \prod_{i=1}^{I} q(y_i|\theta_i = f[x_i, \phi])$$

$$= \arg\min_{\phi} -\sum_{i=1}^{I} \log q(y_i|\theta_i = f[x_i, \phi])$$

Lin. Reg.: Least Squares Loss

Assuming $y_i|x_i \sim N(f[x_i, \phi], \sigma^2)$

$$\hat{\phi} = \underset{\phi}{\arg\min} - \sum_{i=1}^{I} \log q(y_i|\theta_i = f[x_i, \phi])$$

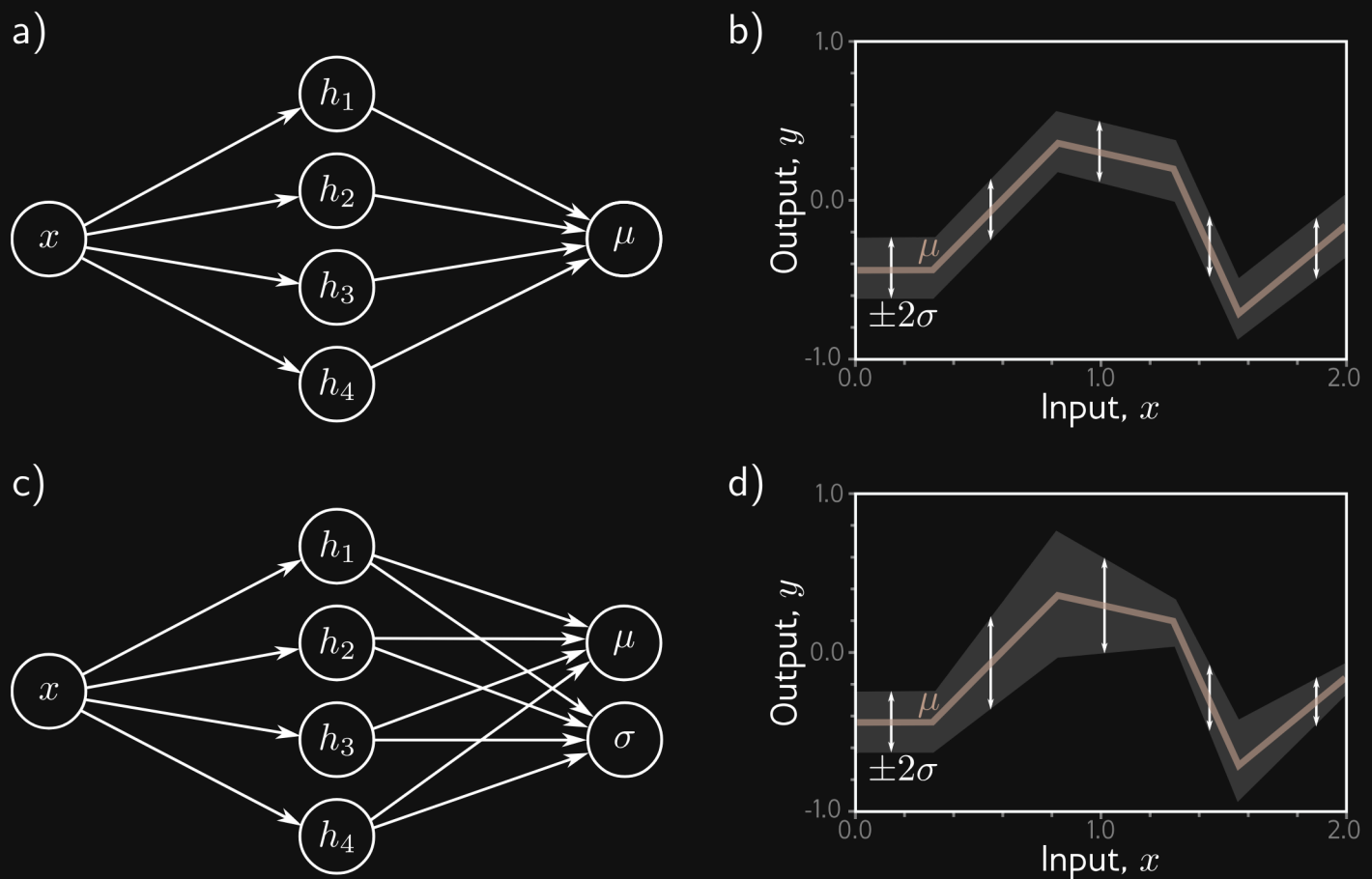
$$= \underset{\phi}{\arg\min} - \sum_{i=1}^{I} \log \left(\frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(y_i - f[x_i, \phi])^2}{2\sigma^2} \right) \right)$$

$$= \underset{\phi}{\arg\min} \sum_{i=1}^{I} - \log \left(\frac{1}{\sqrt{2\pi\sigma^2}} \right) + \frac{(y_i - f[x_i, \phi])^2}{2\sigma^2}$$

$$= \underset{\phi}{\arg\min} \sum_{i=1}^{I} (y_i - f[x_i, \phi])^2$$

Heteroscedastic Regression

$$\arg\min_{\phi} -\sum_{i=1}^{I} \log\left(\frac{1}{\sqrt{2\pi f_2[x_i,\phi]^2}}\exp\left(-\frac{(y_i - f_1[x_i,\phi])^2}{2f_2[x_i,\phi]^2}\right)\right)$$



Binary Classification: Binary Cross-Entropy Loss

Assuming $q(y_i = k|x_i) = softmax_k(f[x_i, \phi])$

$$\hat{\phi} = \arg\min_{\phi} -\sum_{i=1}^{I} \log\left(softmax_{y_i}(f[x_i, \phi])\right)$$

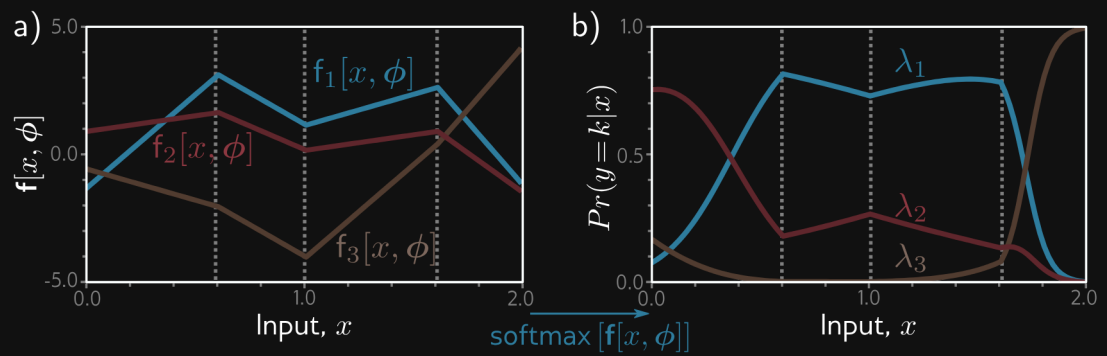
$$= \arg\min_{\phi} -\sum_{i=1}^{I} \left(f_{y_i}[x_i, \phi] - \log\left(\sum_{k'=1}^{K} \exp(f_{k'}[x_i, \phi])\right)\right)$$

Multiclass Classification: Multiclass Cross-Entropy Loss

Assuming $y_i|x_i \sim Ber(\lambda_i = logit(f[x_i, \phi]))$

$$\hat{\phi} = \arg\min_{\phi} -\sum_{i=1}^{I} \log\left((1 - logit(f[x_i, \phi]))^{1-y_i} logit(f[x_i, \phi])^{y_i}\right)$$

$$= \arg\min_{\phi} \sum_{i=1}^{I} -(1 - y_i)\log\left(1 - logit(f[x_i, \phi])\right) - y_i \log\left(logit(f[x_i\right.$$



Multivariate Output

Assuming independent dimensions $q(y|f[x,\phi]) = \prod_d q(y_d|f_d[x,\phi])$

$$\hat{\phi} = \arg\min_{\phi} -\sum_{i=1}^{I} \log\left(\prod_{d} q(y_{i,d}|f_d[x_i,\phi])\right)$$

$$= \arg\min_{\phi} -\sum_{i=1}^{I}\sum_{d} \log\left(q(y_{i,d}|f_d[x_i,\phi])\right)$$

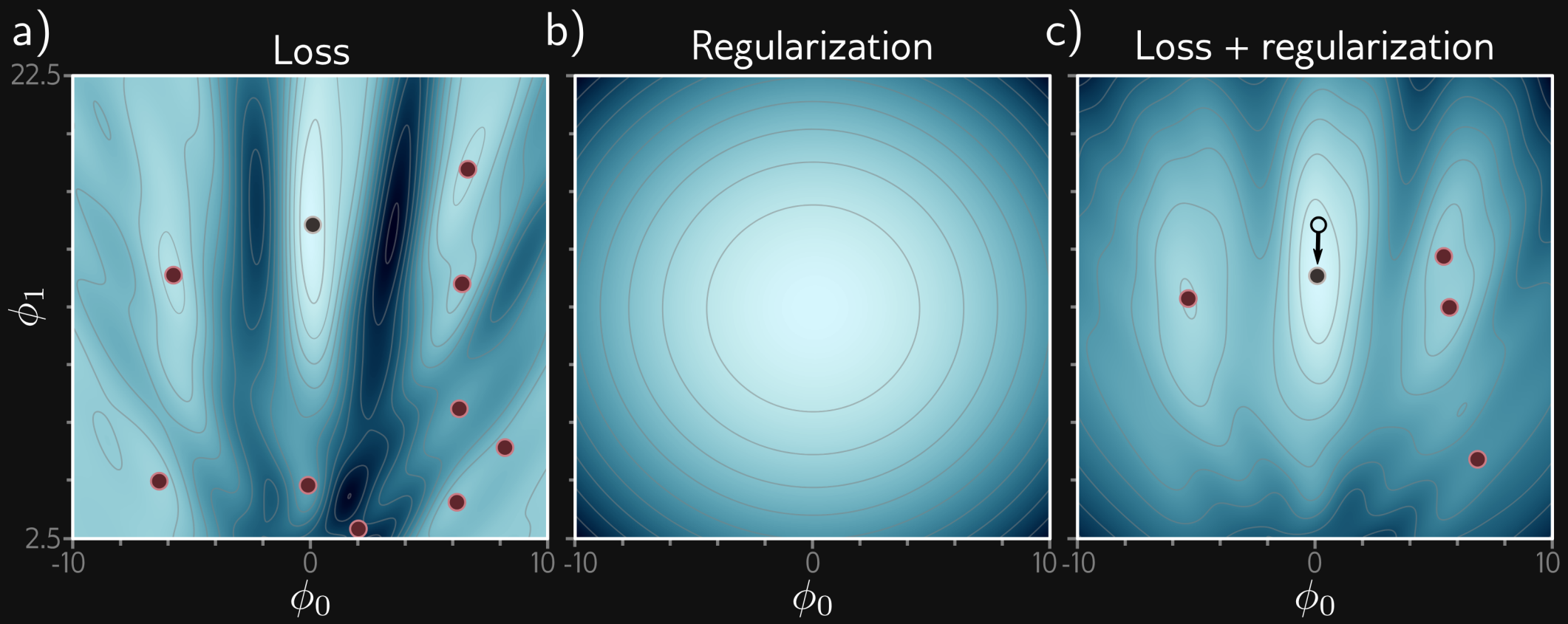
Regularization

Goal: Reduce generalization gap between training and test performance

Prince (2023, chap. 9)

Explicit Regularization

$$\hat{\phi} = \arg\min_{\phi} \sum_{i=1}^{I} l_i[x_i, y_i] + \lambda g[\phi]$$



Interpretation as Bayesian Prior

Maximum a posteriori (MAP) instead of maximum LLH

$$\hat{\phi} = \arg\max_{\phi} \prod_{i=1}^{I} q(y_i|\theta_i = f[x_i, \phi]) P(\phi)$$

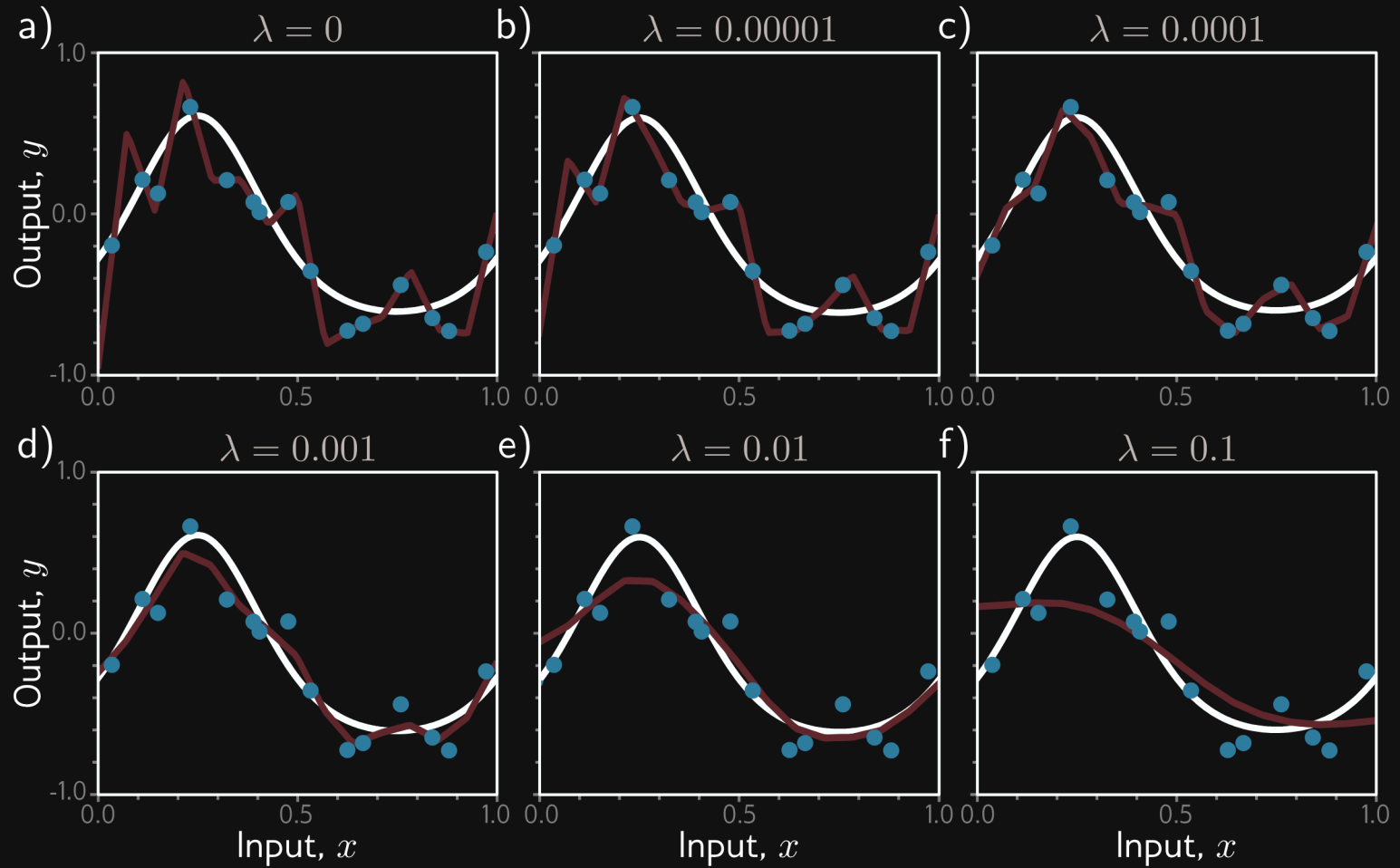
$$= \arg\max_{\phi} \log \prod_{i=1}^{I} q(y_i|\theta_i = f[x_i, \phi]) P(\phi)$$

$$= \arg\min_{\phi} -\log P(\phi) - \sum_{i=1}^{I} \log q(y_i|\theta_i = f[x_i, \phi])$$

$$= \arg\min_{\phi} \lambda g[\phi] + \sum_{i=1}^{I} l_i[x_i, y_i]$$

L2 regularization: Ridge regression

$$g[\phi] = \sum_j \phi_j^2$$



L1 regularization: Lasso regression

$$g[\phi] = \sum_j |\phi_j|$$

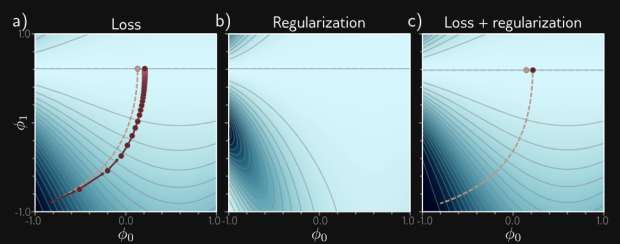
Implicit Regularization

Discrete stochastic gradient descent on L arrives at same place as continuous descent $\frac{d\phi}{dt} = -\frac{\partial \tilde{L}}{\partial \phi}$ on

$$\tilde{L}[\phi] = L[\phi] + \frac{\alpha}{4}||\frac{\partial L}{\partial \phi}||^2 + \frac{\alpha}{4B}\sum_{b=1}^{B}||\frac{\partial L_b}{\partial \phi} - \frac{\partial L}{\partial \phi}||^2$$

so is naturally biased

- away from steep gradients
- to stable gradients (low variance across batches), suggesting that “all data fits well” and promising better generalization



Stefan Bucher

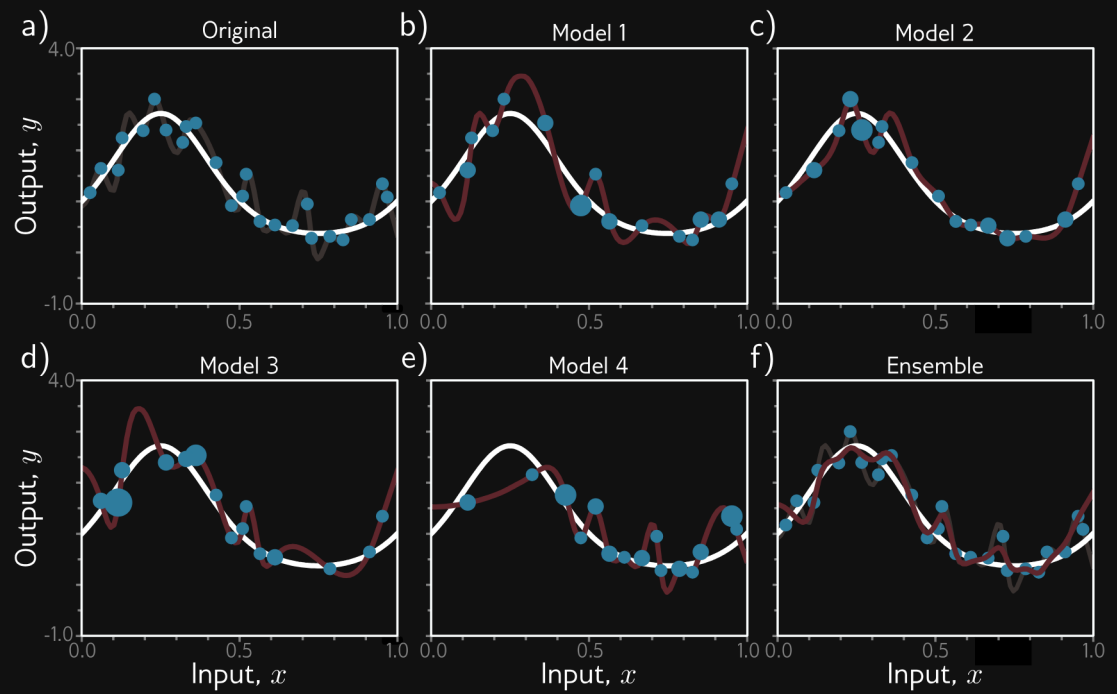
Performance Heuristics

- Early stopping (to avoid overfitting)
- Applying noise (or adversarial) during training (to inputs, weights, or labels) for increased robustness
- Bayesian inference $P(\phi|\{x_i, y_i\}) = \propto \prod_{i=1}^{I} q(y_i|x_i, \phi)P(\phi)$ (often not practical)
- Normalization

Ensemble Methods

Average predictions of multiple models, e.g.

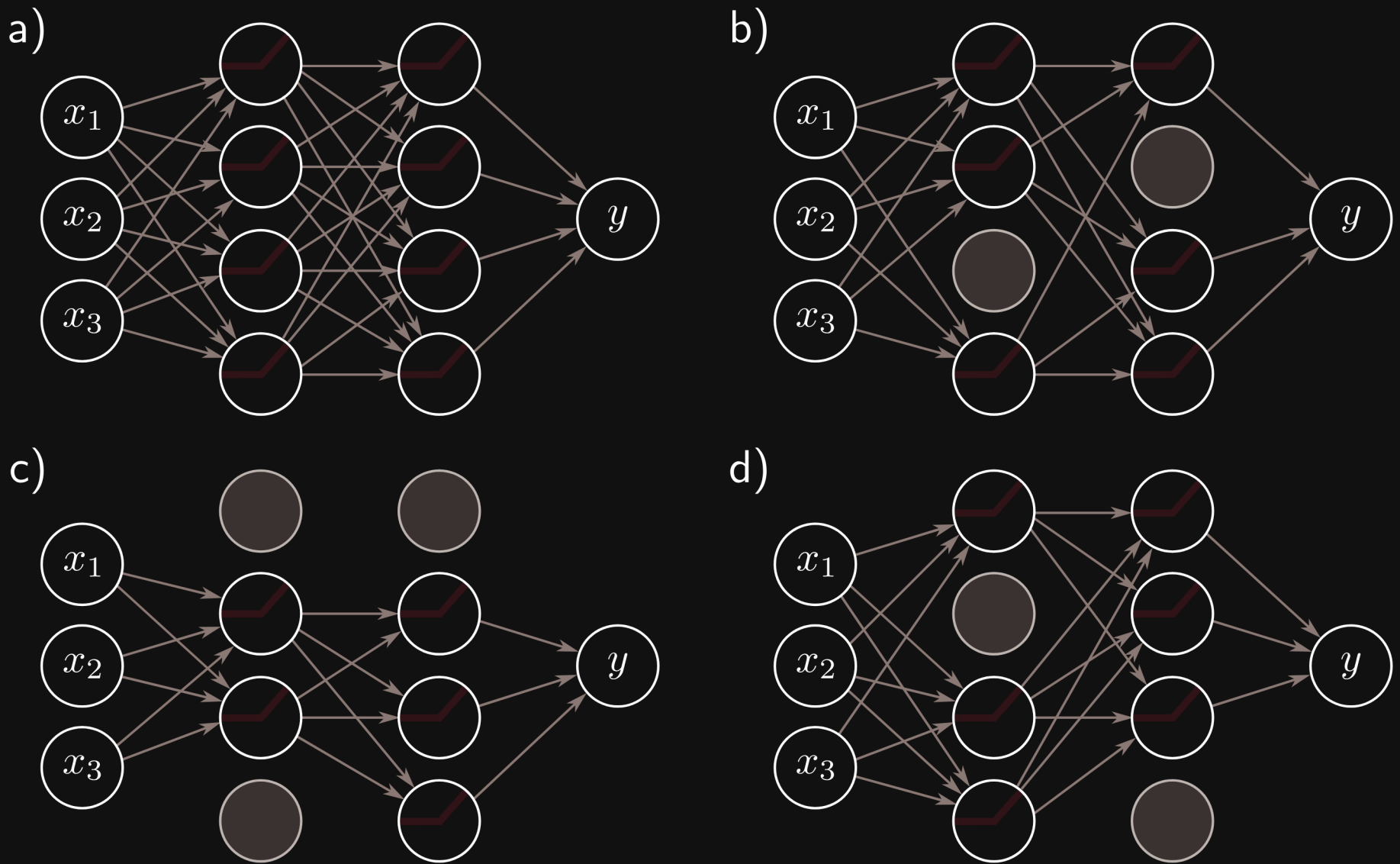
- different initializations
- Bagging (bootstrap aggregating, i.e. resampling training data)



- Boosting (iteratively improve weak learners)

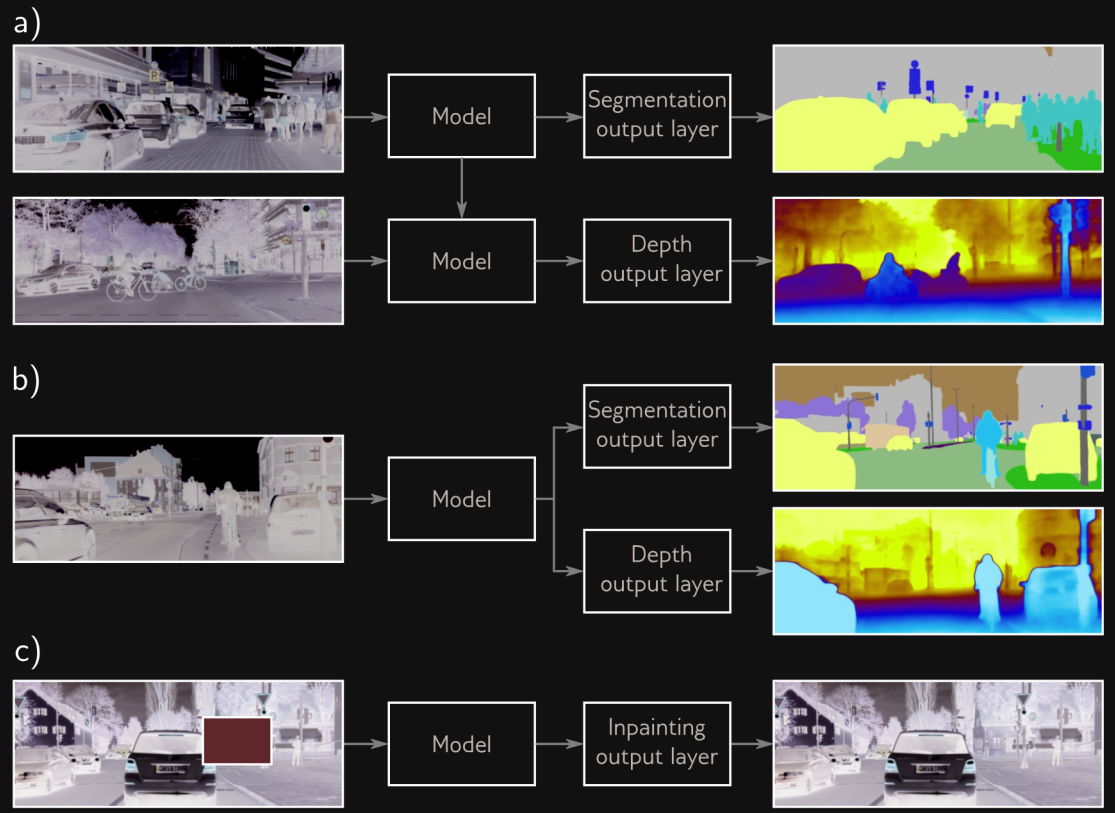
Dropout

Clamping subset of hidden units to zero.

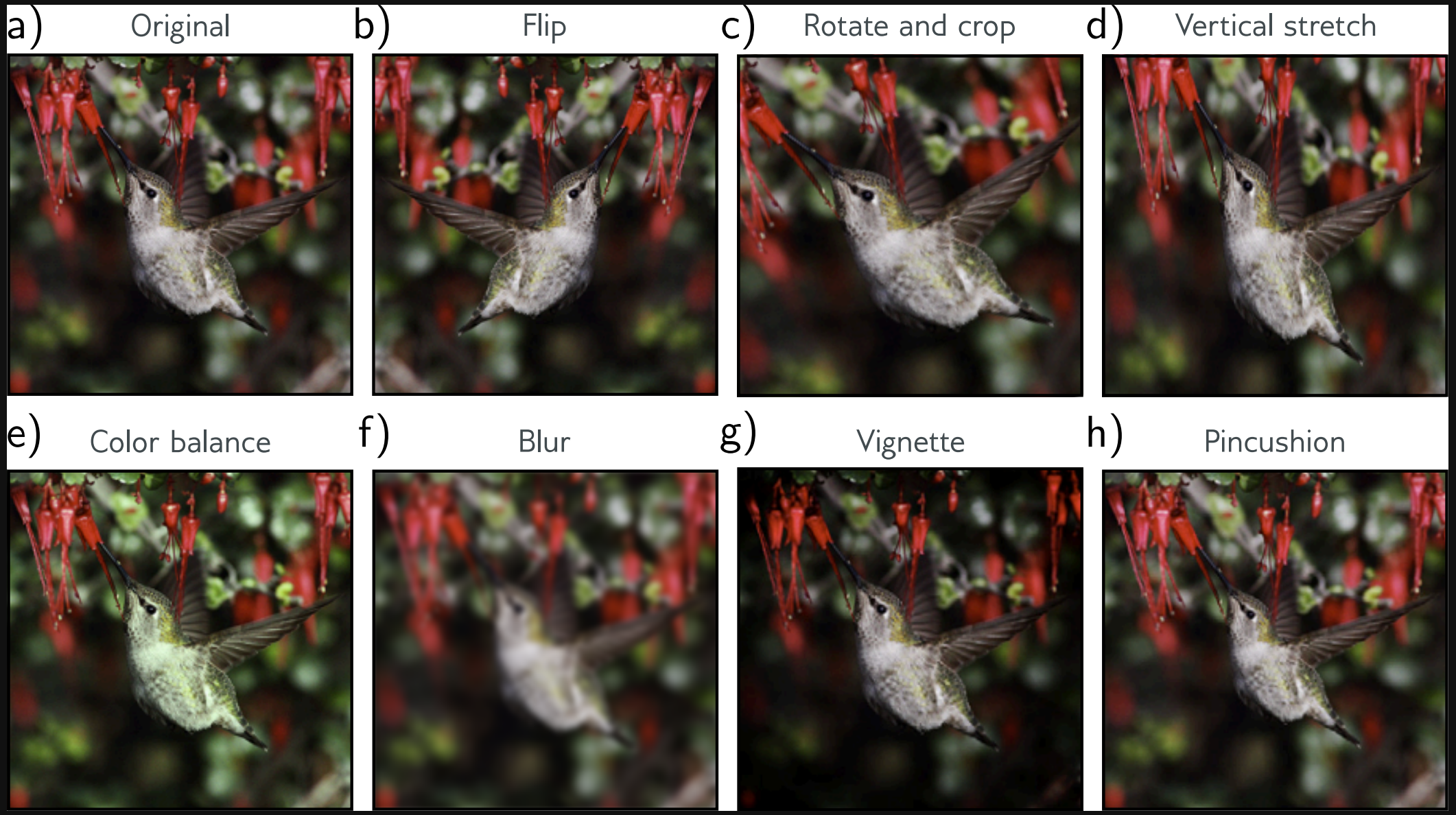


Transfer, Multi-task & Self-supervised Learning

- Transfer learning (pretrain on related training data)
  - train final layers
  - fine-tune whole model
- Multi-task learning
- Self-supervised learning
  - generative (fill in masked data)
  - contrastive (compare pairs of examples for relatedness)


a)
Model
Segmentation output layer
Model
Depth output layer
b)
Model
Segmentation output layer
Depth output layer
c)
Model
Inpainting output layer

Data Augmentation



Regularization Overview