Bayesian neural nets

Estimating a predictive distribution

Yarin Gal & Zoubin Ghahramani (2016)
Dropout as a bayesian approximation: Representing model uncertainty in deep learning

Why to care about uncertainty

- Cat / dog classifier, classify an hippopotamus
- Reliability in steering control
- Physics simulator prediction
- Minimising action randomness when connected to a reward

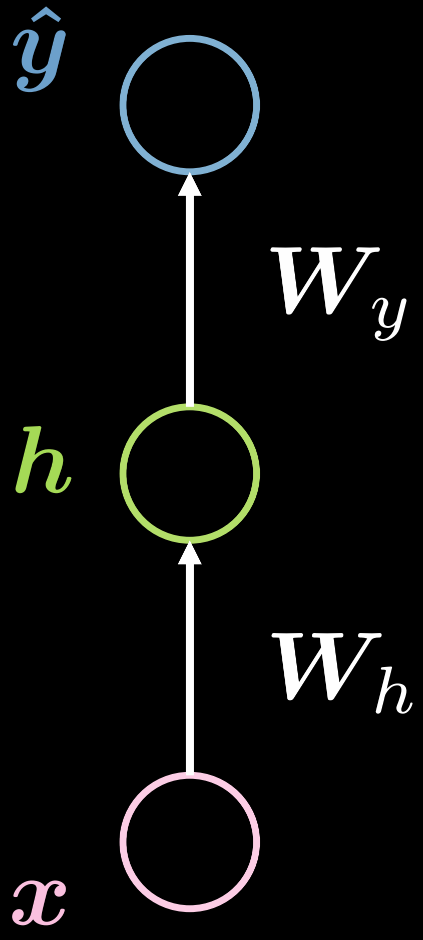
NN with dropout

$$\boldsymbol{h} = f(\boldsymbol{W}_h \boldsymbol{x} \odot \boldsymbol{\delta}_x + \boldsymbol{b}_h)$$

$$\hat{\boldsymbol{y}} = g(\boldsymbol{W}_y \boldsymbol{h} \odot \boldsymbol{\delta}_h + \boldsymbol{b}_y)$$

$$\boldsymbol{x}, \boldsymbol{\delta}_x \in \mathbb{R}^n$$

$$\boldsymbol{h}, \boldsymbol{\delta}_h \in \mathbb{R}^d$$



$$(\boldsymbol{\delta}_x)_i, (\boldsymbol{\delta}_h)_j \sim \mathrm{Ber}(1 - r)$$

$$\hat{\boldsymbol{y}} \in \mathbb{R}^K$$

$$\boldsymbol{W}_h \in \mathbb{R}^{d \times n}$$

$$\boldsymbol{\delta} \leftarrow \frac{1}{1 - r} \boldsymbol{\delta}$$

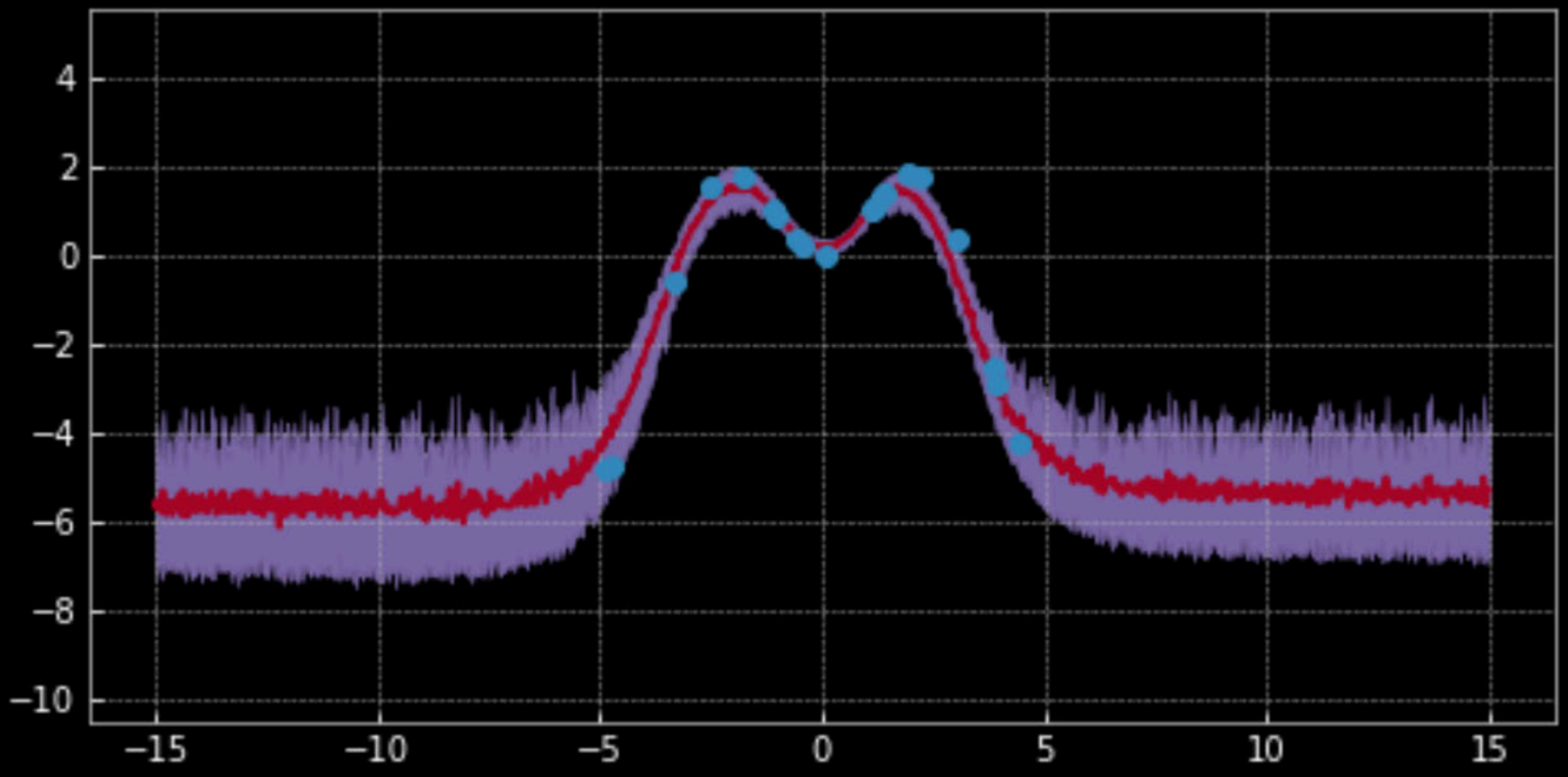
dropping rate

$$\boldsymbol{W}_y \in \mathbb{R}^{K \times d}$$

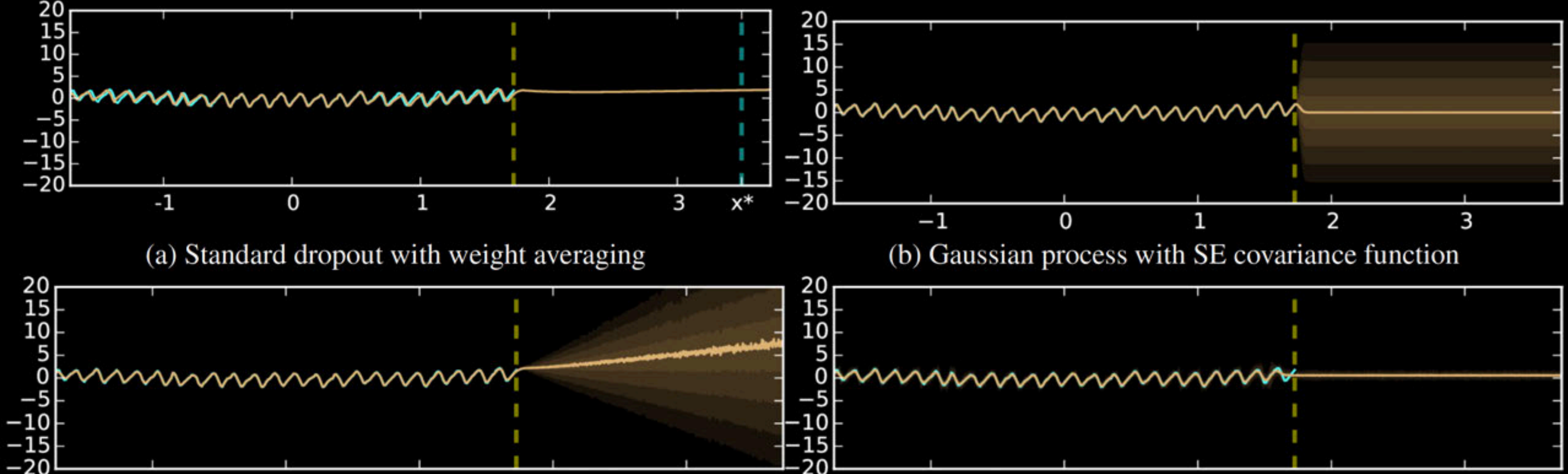
$$f(\cdot), g(\cdot) : (\cdot)^+, \sigma(\cdot), \tanh(\cdot), \mathrm{softmax}(\cdot)$$

Regression (I)

- See notebook demo



Regression (II)



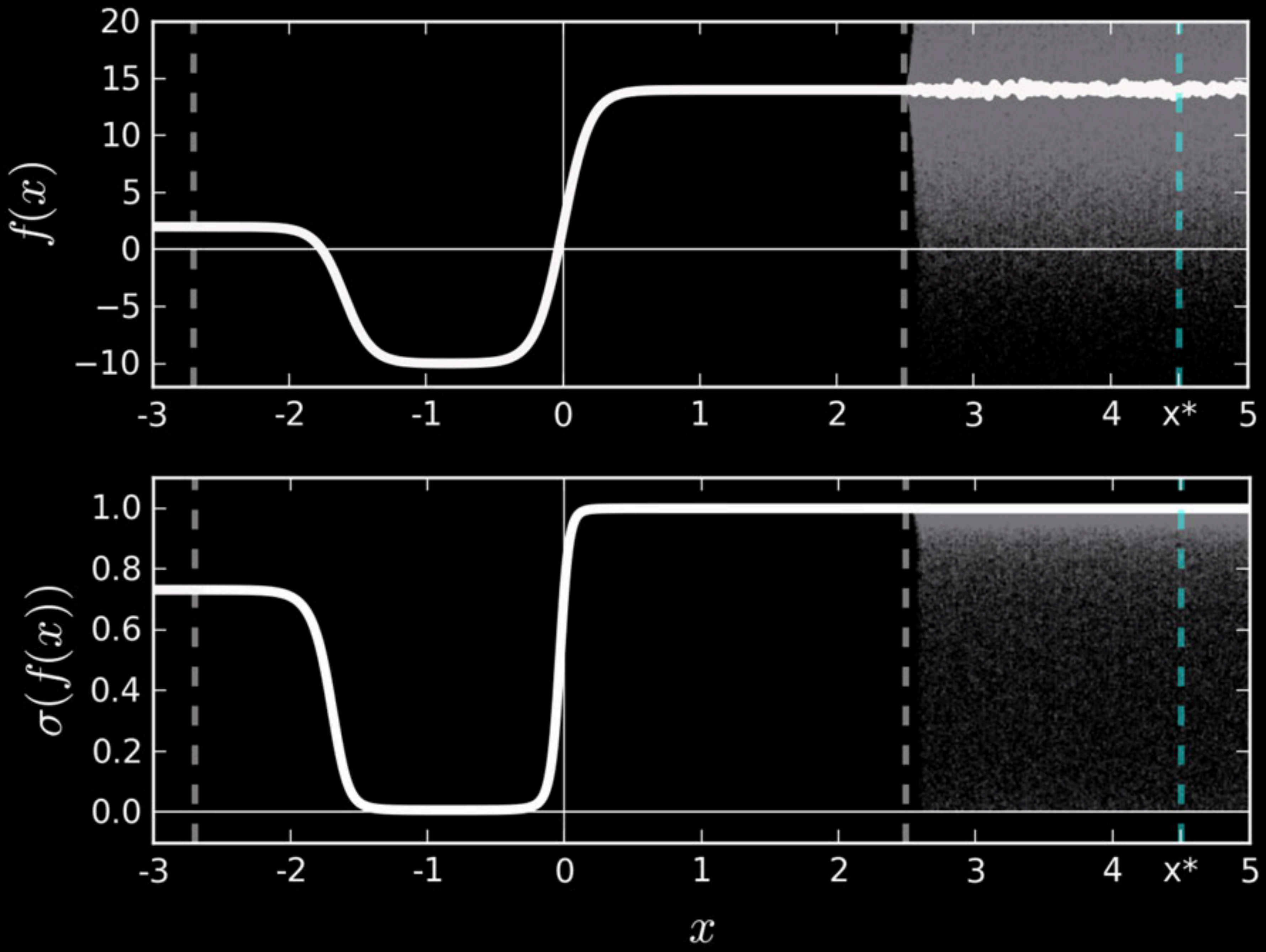
(a) Standard dropout with weight averaging

(b) Gaussian process with SE covariance function

(c) MC dropout with ReLU non-linearities

(d) MC dropout with TanH non-linearities

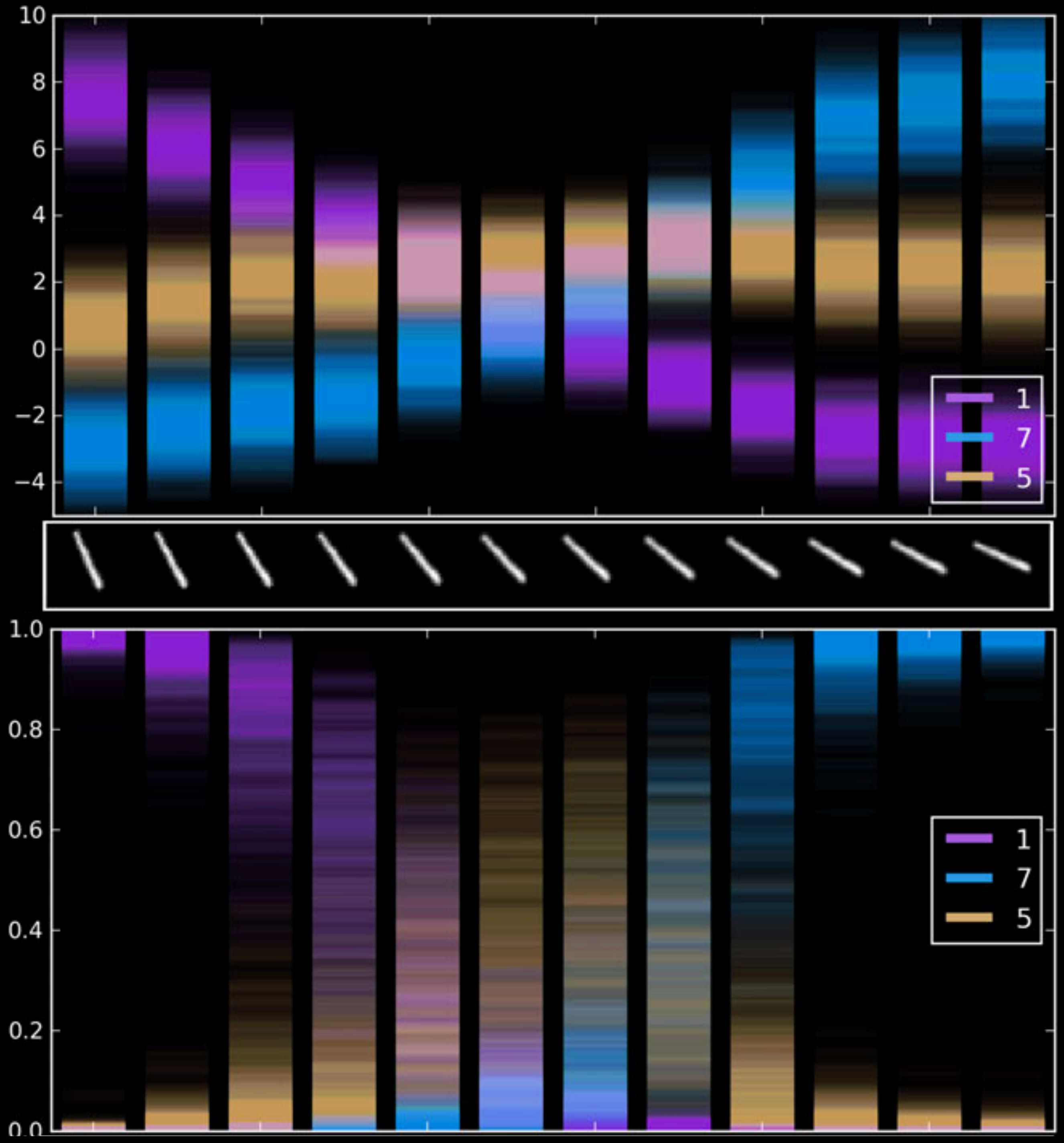
Binary classification



logit →

sigmoid →

Multi-class classification


10
8
6
4
2
0
−2
−4
1
7
5
1.0
0.8
0.6
0.4
0.2
0.0
1
7
5