Zurich University of Applied Sciences

Uncertainty in DL models

Beate Sick

sick@zhaw.ch

Remark: Much of the material has been developed together with Elvis Murina and Oliver Dürr

Topics

- **Capturing uncertainty in statistics**
  - A conditional probability distribution is modeled
  - Parameter of the CPD have uncertainty
  - Spread of CPD quantifies data variation
- **Capturing uncertainty in DL?**
  - Quantifying prediction uncertain via Dropout
  - Calibration

Recap: Linear regression in statistics

Model for the conditional probability distribution

CPD: $Y_{X_i} = (Y|X_i) \sim N(\mu_{x_i}, \sigma^2)$

$Y_x \in \mathbb{R}$

$\mu_x \in \mathbb{R}$

μ_x is modeled

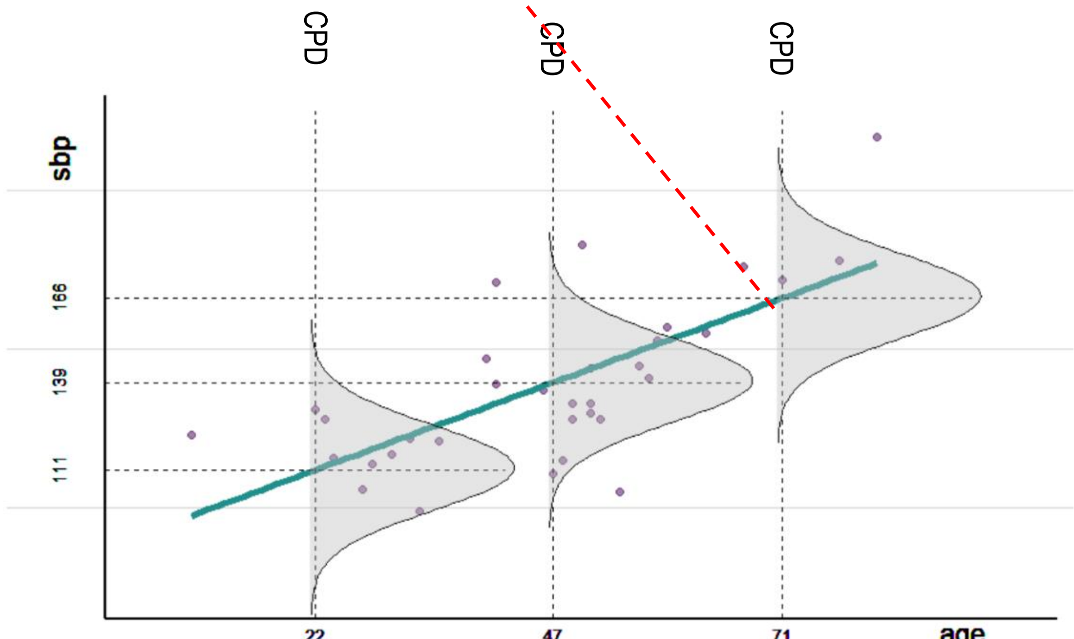
σ^2 is independent of the predictor values

$y_i = \beta_0 + \beta_1 \cdot x_{i1} + \varepsilon_i$

$E(Y_{X_i}) = \mu_{x_i} = (\mu | X = x_i) = \beta_0 + \beta_1 \cdot x_{i1}$

$Var(Y_{X_i}) = Var(Y|X_i) = Var(\varepsilon_i) = \sigma^2$

ε_i i.i.d. $\sim N(0, \sigma^2)$



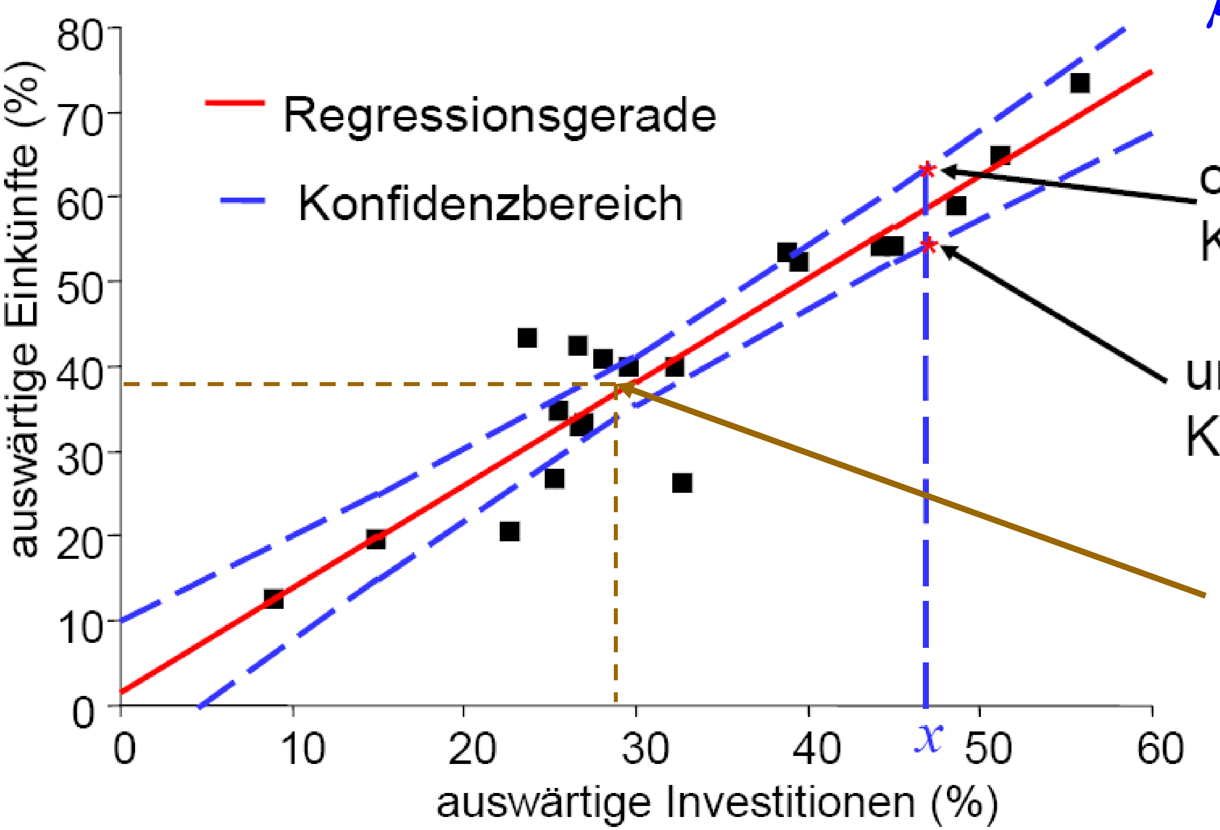
Confidence Interval

Model for the conditional probability distribution

CPD: $Y_{X_i} = (Y|X_i) \sim N(\mu_{x_i}, \sigma^2)$

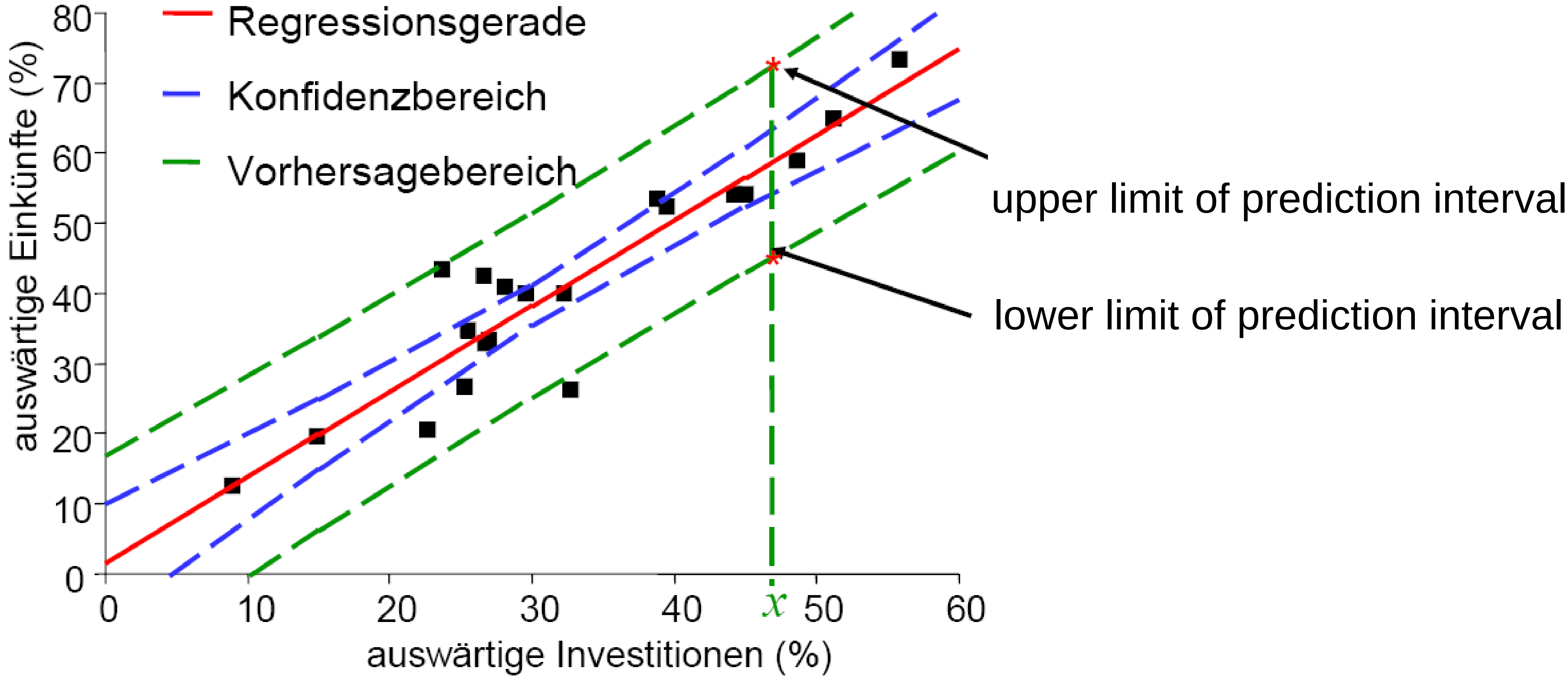
We can determine at each position x_i the confidence interval for μ_{x_i}

$$\hat{\mu}_{x_i} \pm q^{t_{n-2}}_{\alpha/2} \cdot se(\hat{\mu}_{x_i})$$



The fitted regression line runs always through the center of the data

Confidence and Prediction Interval: Visualization



The confidence interval quantifies the uncertainty of the parameter μ ("aleatoric uncertainty")

The prediction interval quantifies the variation of the data (" epistemic uncertainty")

Confidence and Prediction Interval: Formula

$$Y_i = a + bX_i + \varepsilon_i$$

CI for μ

$$\hat{\mu}_x \pm q^{t_{n-2}}{}_{\alpha/2} \cdot se(\hat{\mu}_x)$$

$$se(\hat{\mu}_x) = \sqrt{\hat{\sigma}^2 \cdot \left(\frac{1}{n} + \frac{(x-\overline{x})^2}{\sum (x_i - \overline{x})^2}\right)}$$

CI for σ

$$\hat{\sigma} = \sqrt{\frac{1}{n-2}\sum_{i=1}^{n} r^2{}_i}$$

$$\hat{\sigma} \pm q^{t_{n-2}}{}_{\alpha/2} \cdot se(\hat{\sigma})$$

$$se(\hat{\sigma}) = \sqrt{\hat{\sigma}^2 \cdot \left(1 + \frac{1}{n} + \frac{(x-\overline{x})^2}{\sum (x_i - \overline{x})^2}\right)}$$

PI for y

$$\hat{\mu}_x \pm q^{t_{n-2}}{}_{\alpha/2} \cdot se(y_x)$$

$$se(y_x)^2 = se(\hat{\mu}_x)^2 + \hat{\sigma}^2$$

Recap: Binary classification in statistics

CPD: $Y_{X_i} = (Y|X_i) \sim \mathrm{Ber}(p_{x_i})$

$Y_x \in \{0,1\}$, $p_x \in [0,1]$

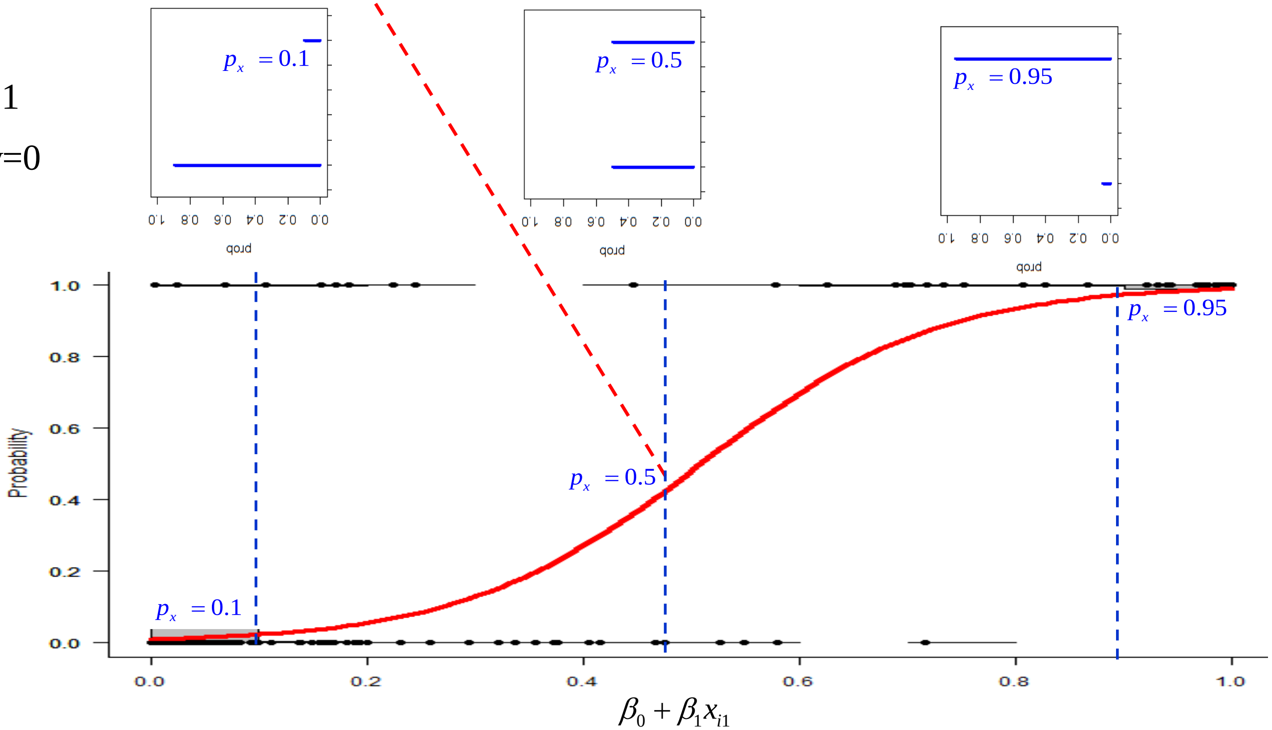
$$\log\left(\frac{p_{x_i}}{1-p_{x_i}}\right) = \beta_0 + \beta_1 x_{i1}$$

$$E(Y_{x_i}) = E(Y=1 \mid X = x_i) = p_{x_i} = \frac{e^{\beta_0+\beta_1 x_{i1}}}{1+e^{\beta_0+\beta_1 x_{i1}}}$$

$$\mathrm{Var}(Y_{x_i}) = p_{x_i} \cdot (1-p_{x_i})$$

p_x is modeled

$$P(\mathrm{Y} \mid \mathrm{X}=\mathrm{x}) = \begin{cases} p_x & , \mathrm{y}=1 \\ 1-p_x & , \mathrm{y}=0 \end{cases}$$

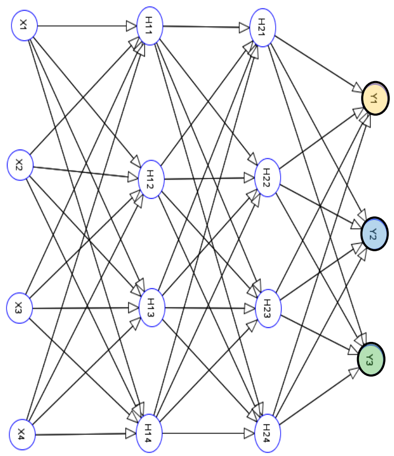


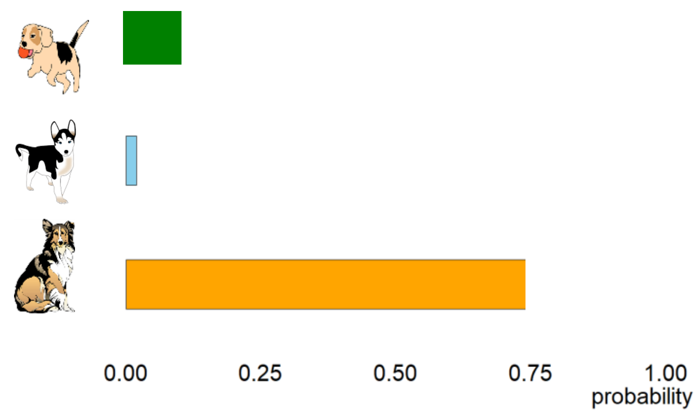
DL can also be used for regression and classification, but…

Can DL quantify uncertainty?

What do we get from a DL classification model?

Suppose you train a classifier to discriminate between 3 dog breeds





The prediction is “collie” because it gets the highest probability: p_{max}=0.75

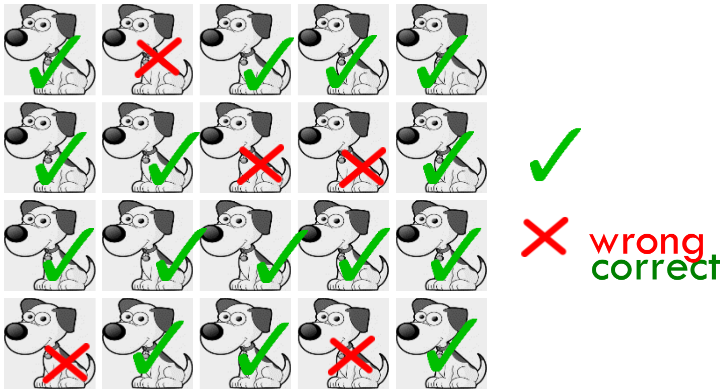
What is the probability telling?

p_{max}=0.75

Among many predications that had p_{max}=0.75, we expect that on average 75% of these predictions are correct and only 25% predictions are wrong

→ Then the classifier produces *calibrated probabilities*

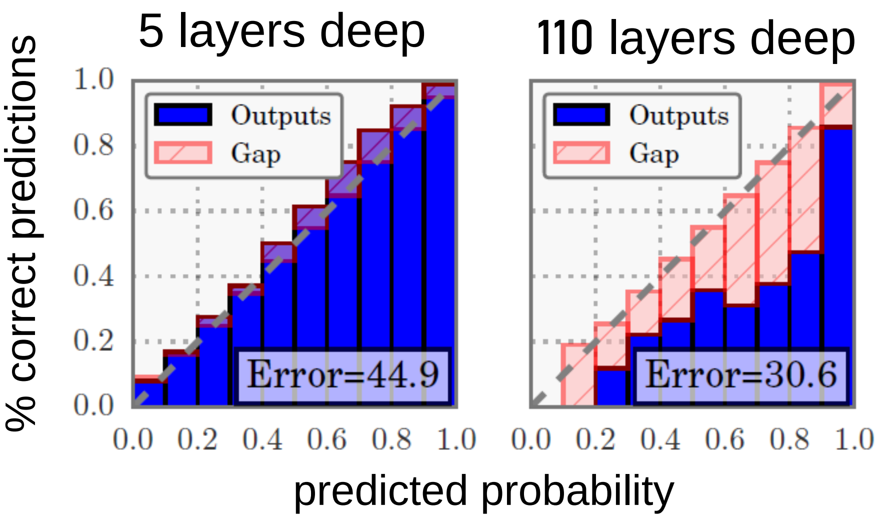
Sample of images where the predicted class got p=0.75:



Do CNNs produce calibrated probabilities?

Guo et al. (2017)
[On Calibration of Modern NN]

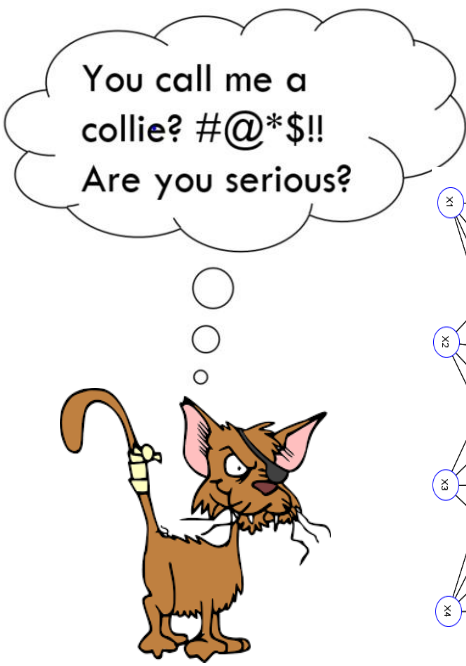
The deeper CNNs get
- the fewer miss-classifications
- the less well calibrated they get

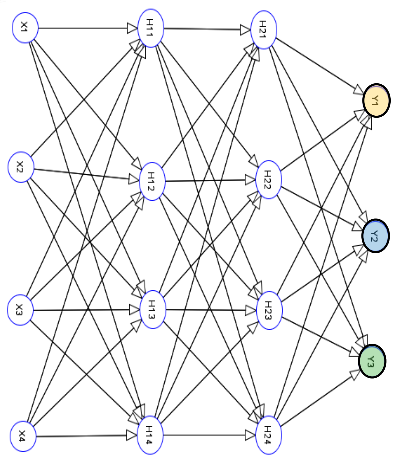


Good news:
deep NN can be "recalibrated" and then we get calibrated probabilities.

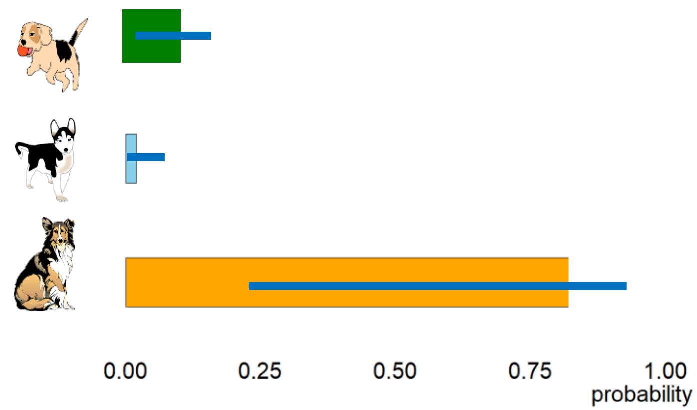
CNNs yield high accuracy
and calibrated probabilities, but…

What happens if a novel class is presented to the CNN?



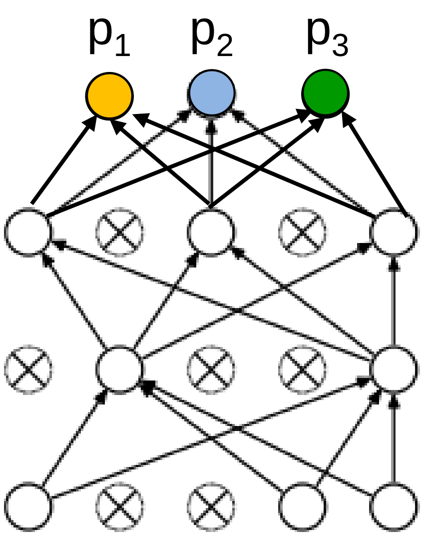


Plain wrong !



We need some error bars!

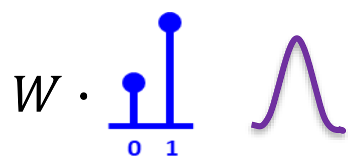
MC Dropout and Bayesian Neural Networks

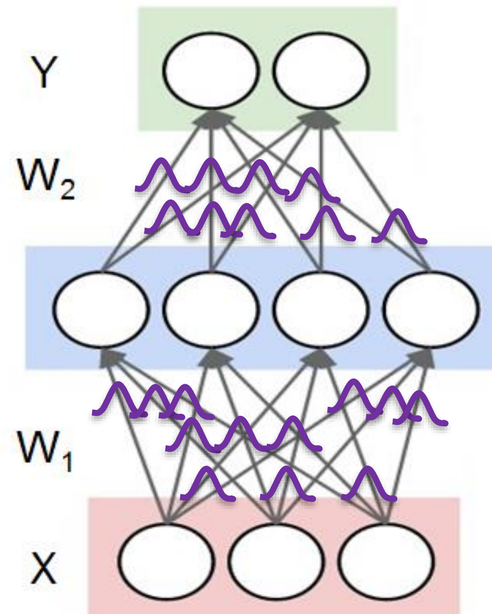


MC Dropout
Randomly drop nodes in each run
→ Ususally done during training

Dropout in test time

Yarin Gal* (2015): we learn a whole weight distribution



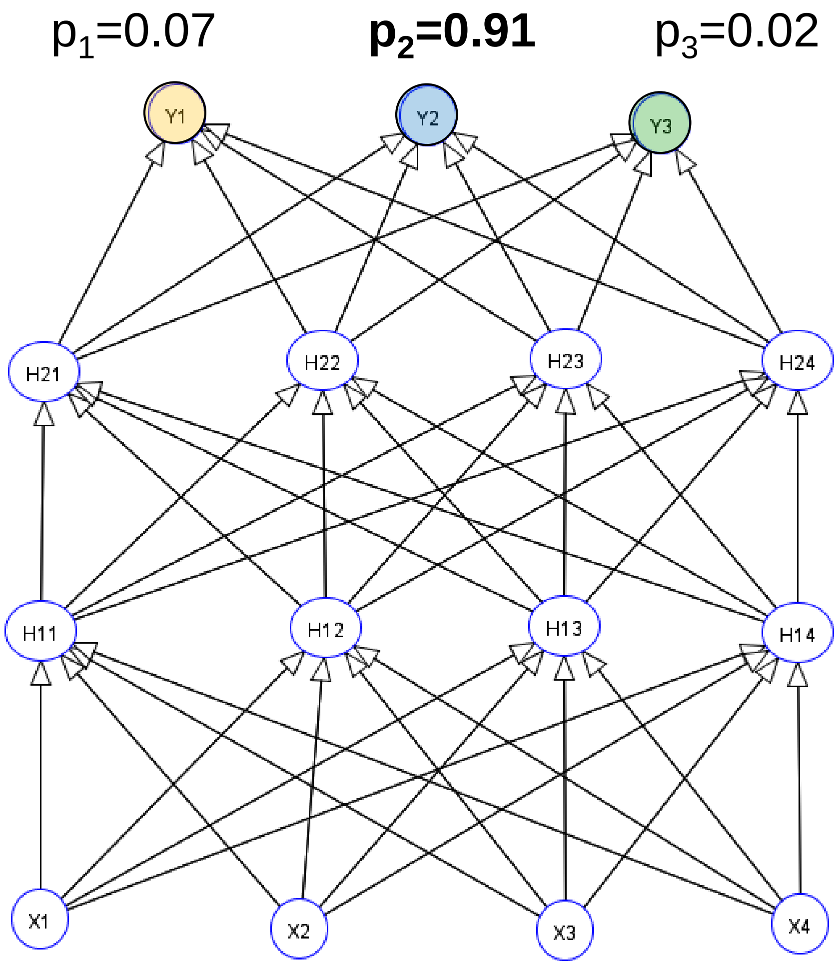


Bayesian NN

→ We should sample from weight distribution during test time

*Dropout as a Bayesian Approximation: Representing Model Uncertainty in Deep Learning https://arxiv.org/abs/1506.02142

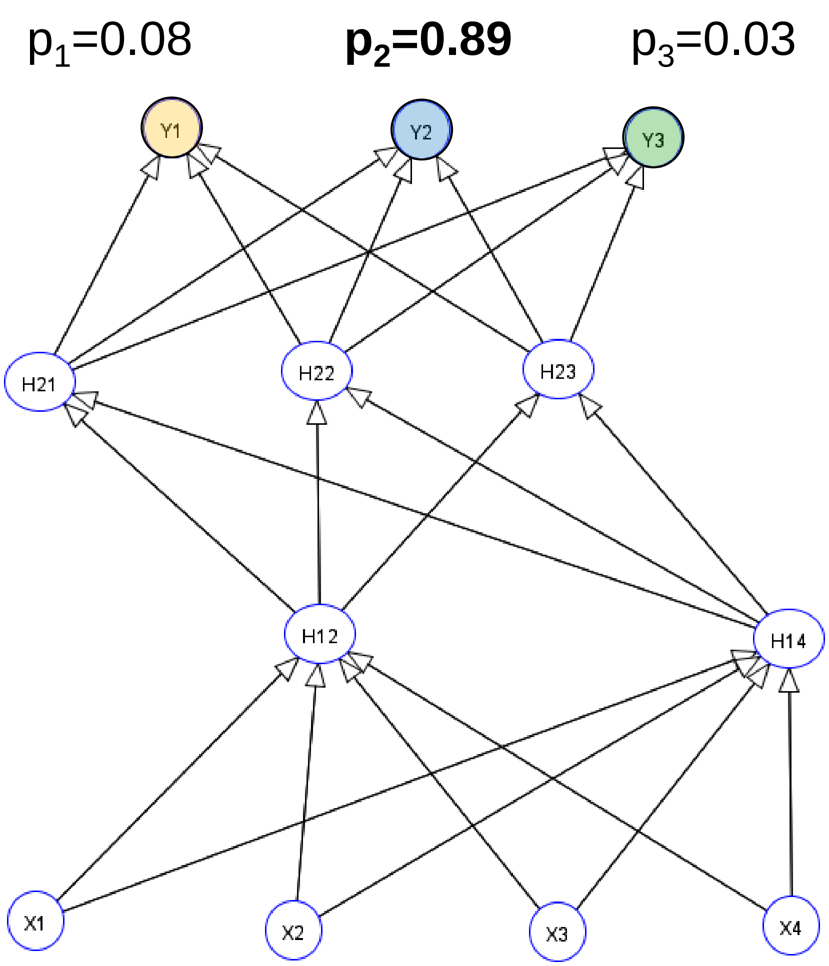
No MC Dropout

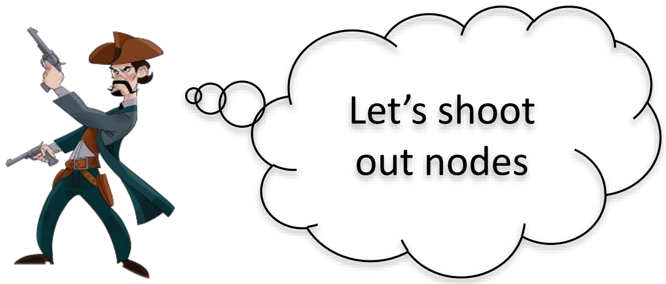


Probability of predicted class: $\mathbf{p_{max}}$

Input: image pixel values

Use dropout at test time: Run 1



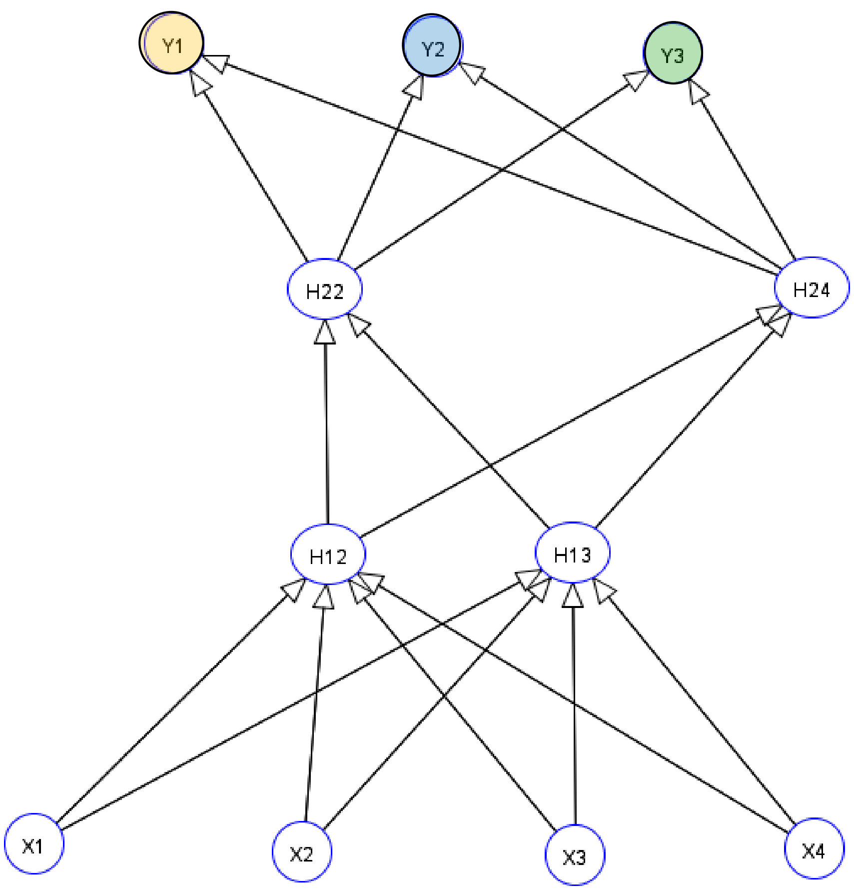


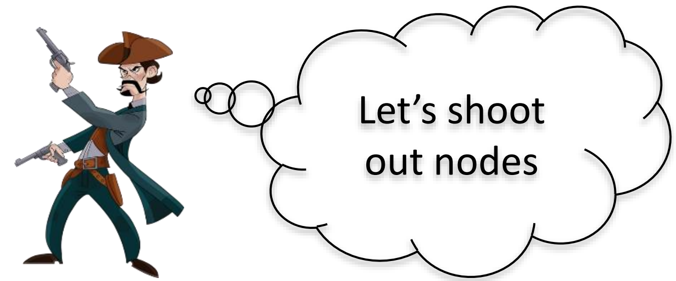
Stochastic dropout of units

Same input image

Use dropout at test time: Run 2

p_1=0.11 **p_2=0.81** p_3=0.08



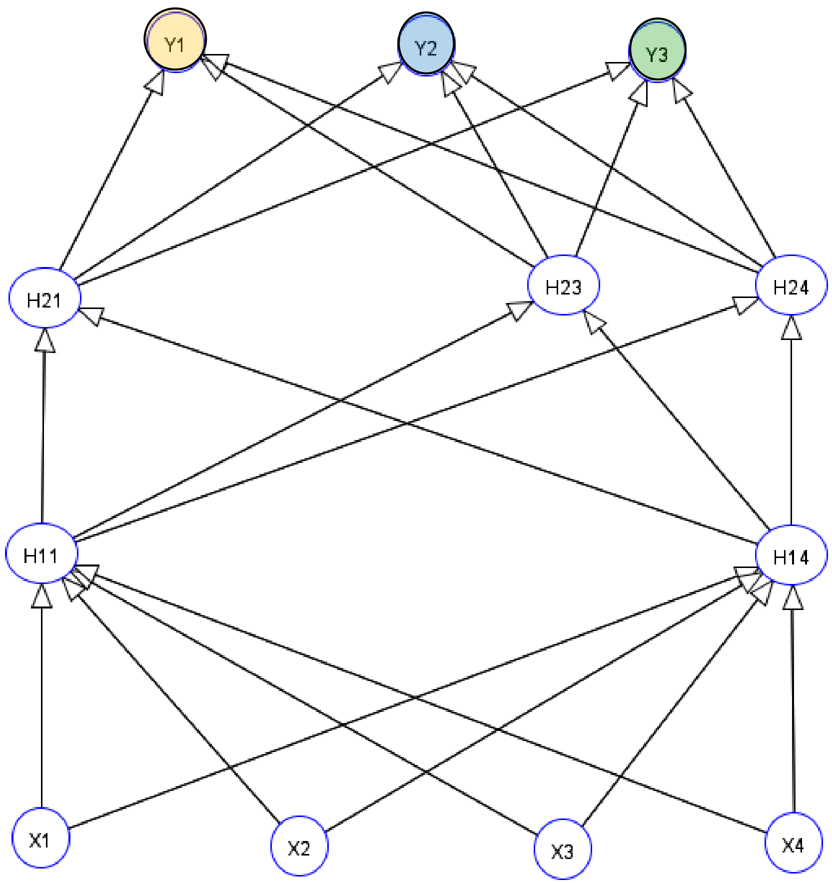


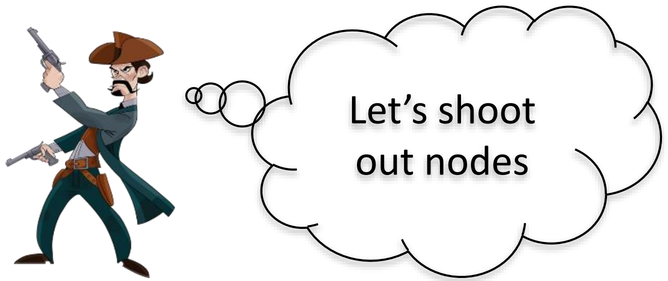
Stochastic dropout of units

Same input image

Use dropout at test time: Run 3

p_1=0.03 **p_2=0.94** p_3=0.03



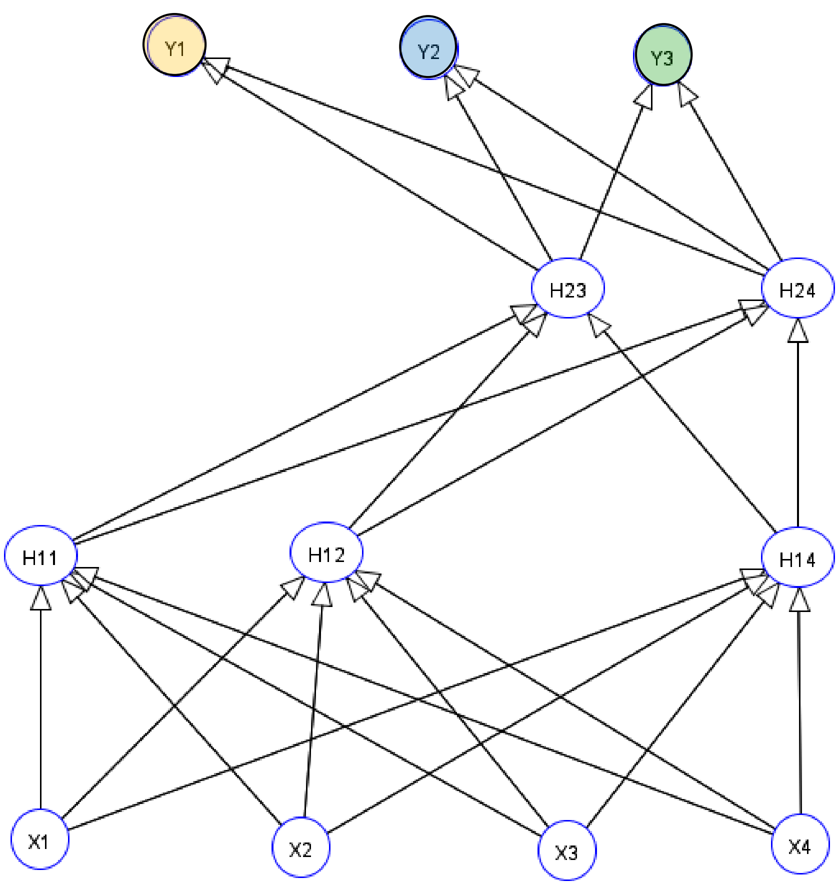


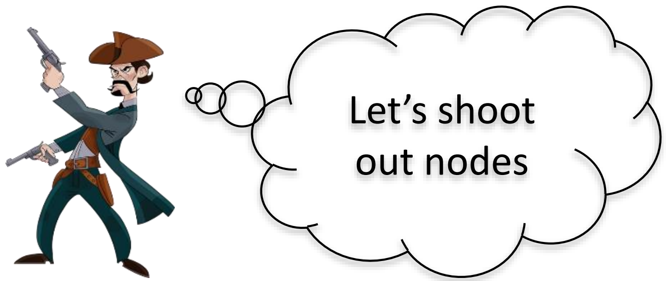
Stochastic dropout of units

Same input image

Use dropout at test time: Run 4

p_1=0.16 **p_2=0.78** p_3=0.06



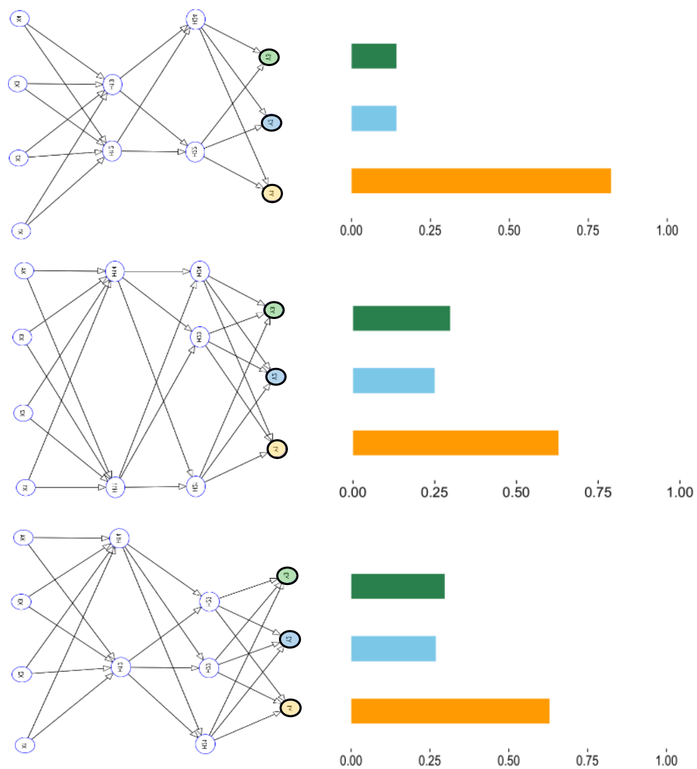


Stochastic dropout of units

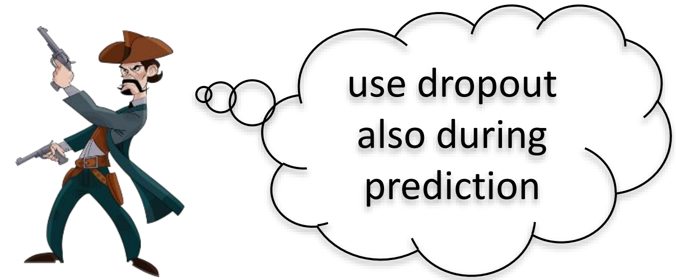
Same input image

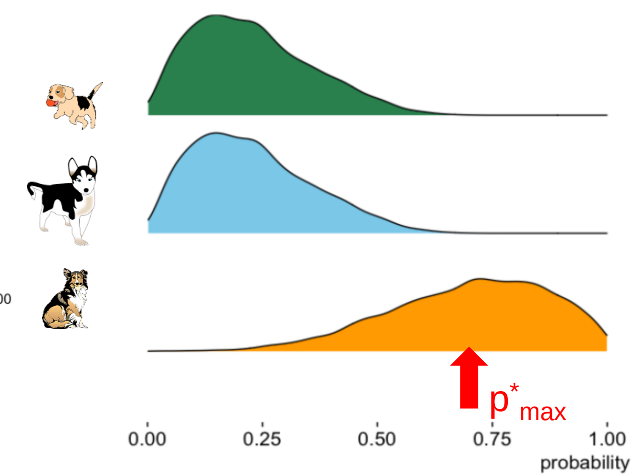
MC probability prediction

Many Dropout Runs in forward pass



...





CNN predicts class
“collie”
but with high uncertainty

Remark: Mean of marginal give components of mean in multivariate distribution.

What to get from the MC* probability distribution

The center of mass quatifies the predicted probablilty

p^{*}_{max}

The spread quantifies in addition the uncertainty of the predicted probability

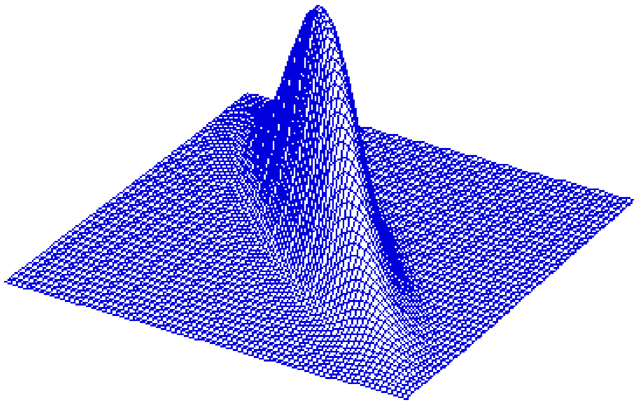
σ^{*} total standard deviation

PE* entropy,

MI* mutual information

VR* variation ratio

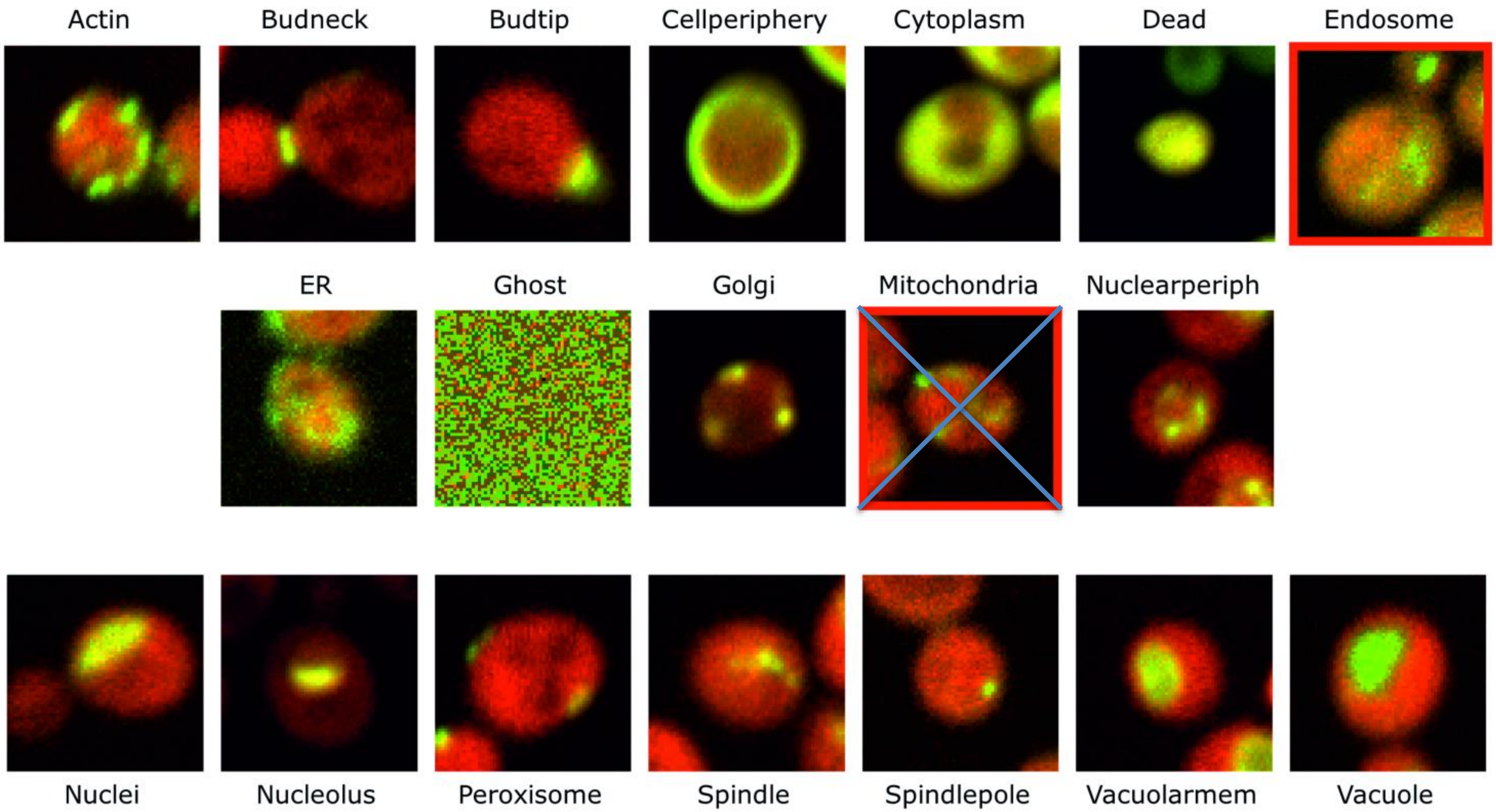
f* vote ratio



Remark: to go from image to patient level we use the average of all measures as baseline

Evaluate method on some real data

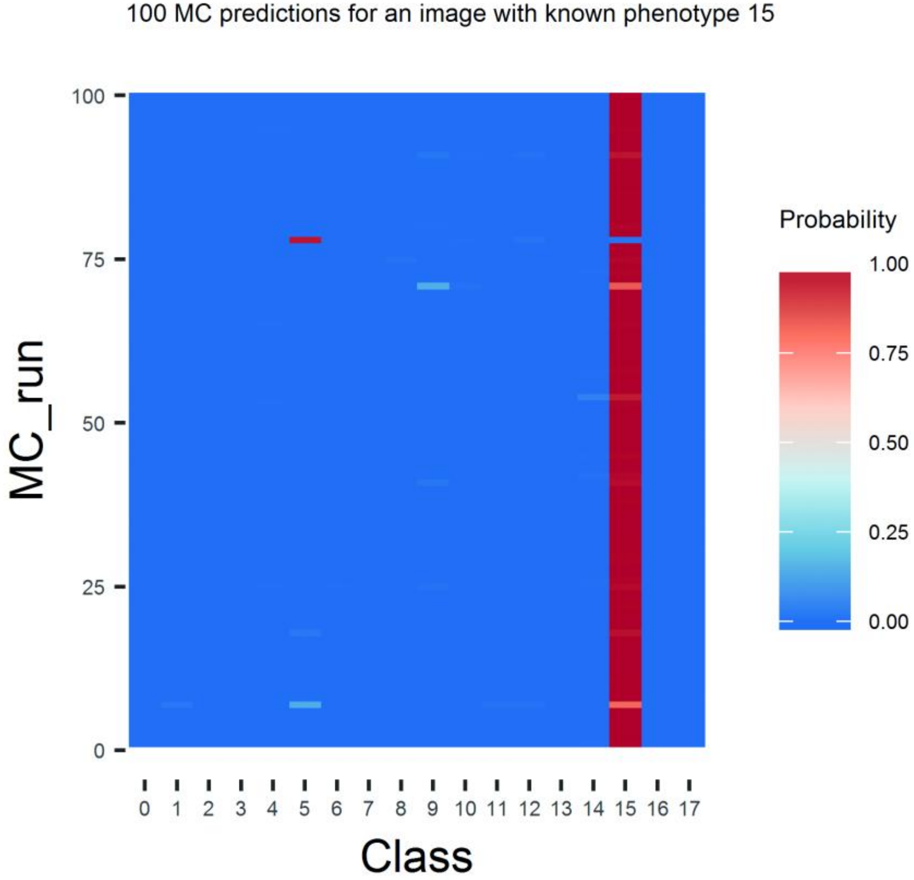
Experiment with unknown phenotype



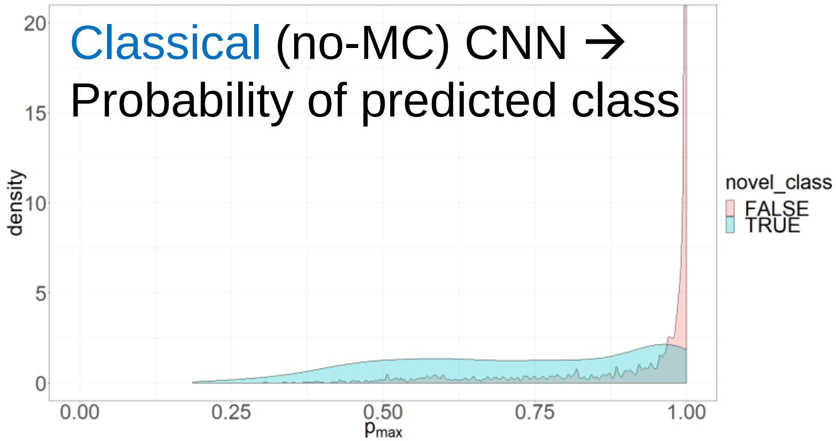
Dürr O, Murina E, Siegismund D, Tolkachev V, Steigele S, Sick B. Know when you don't know, Assay Drug Dev Technol. 2018

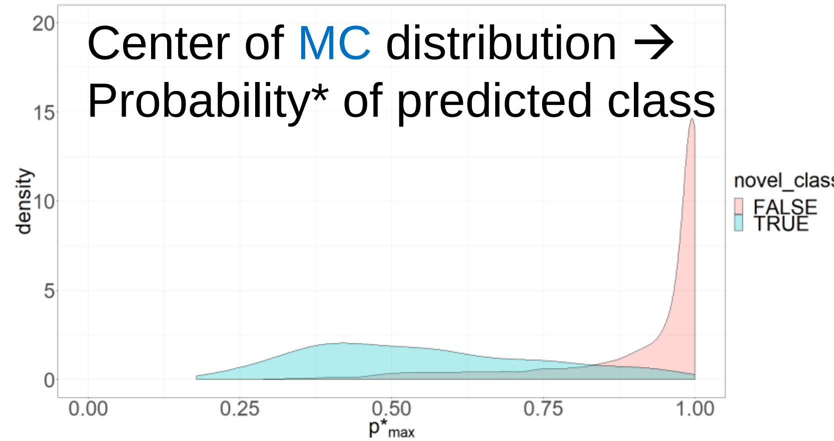
Probability distribution from MC dropout runs

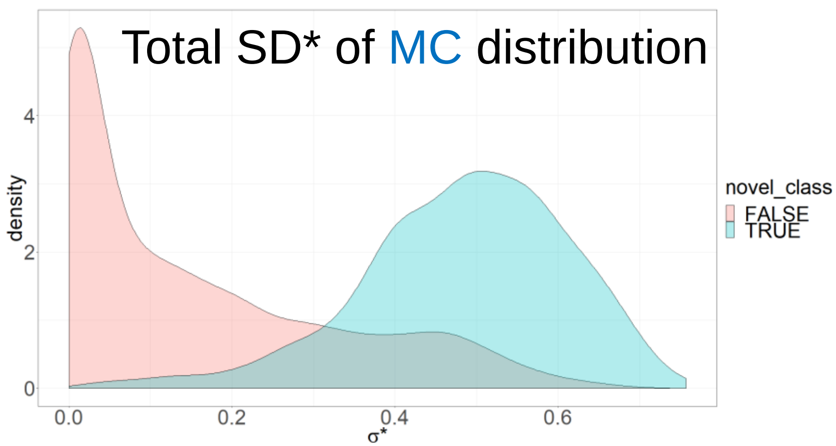
Image with known class 15

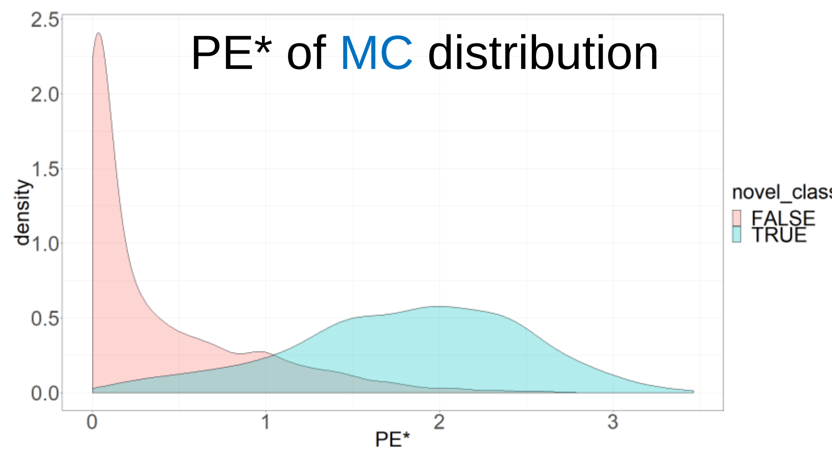


Do known/novel classes yield different values for probability and uncertainty measures?

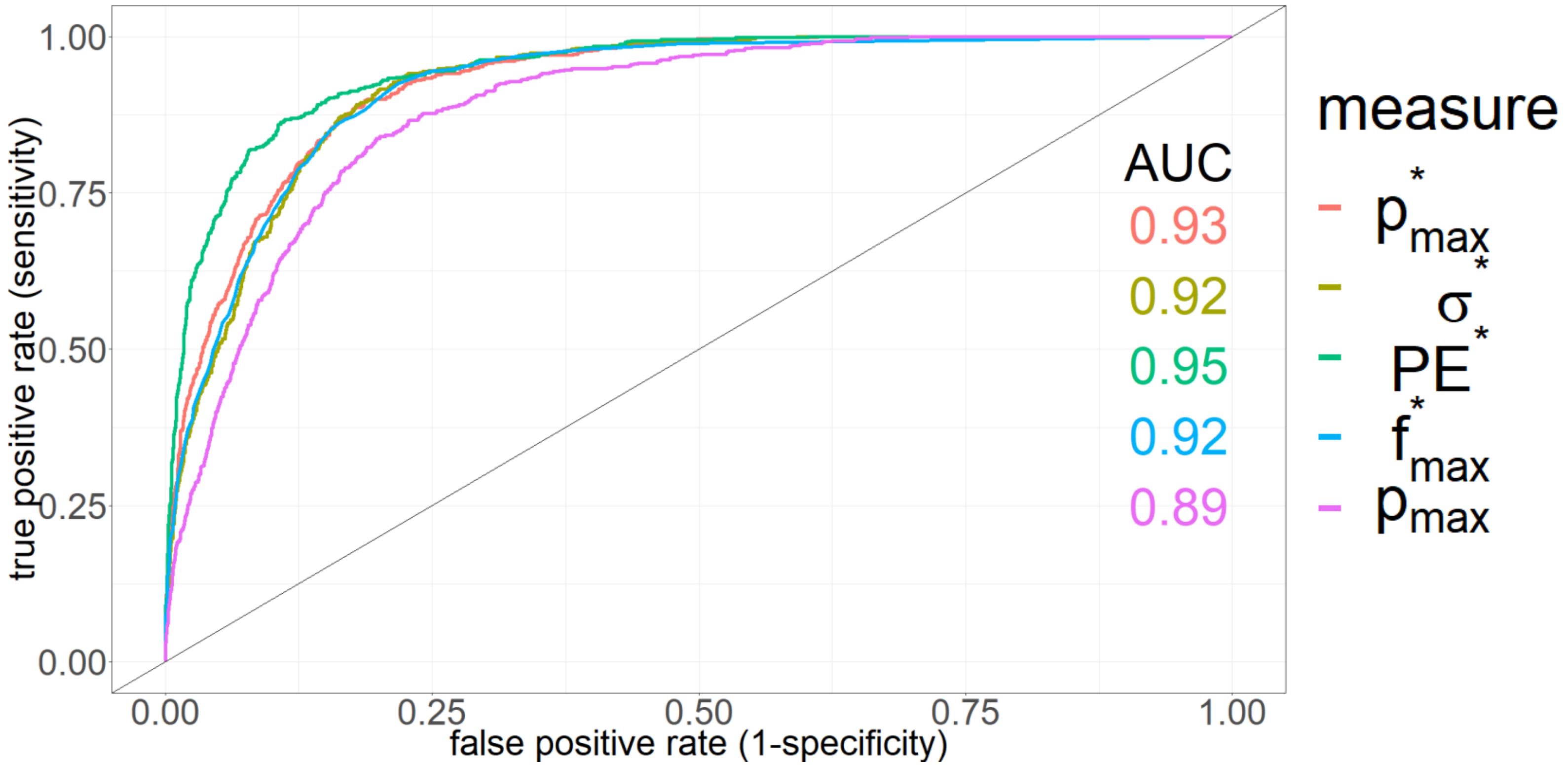








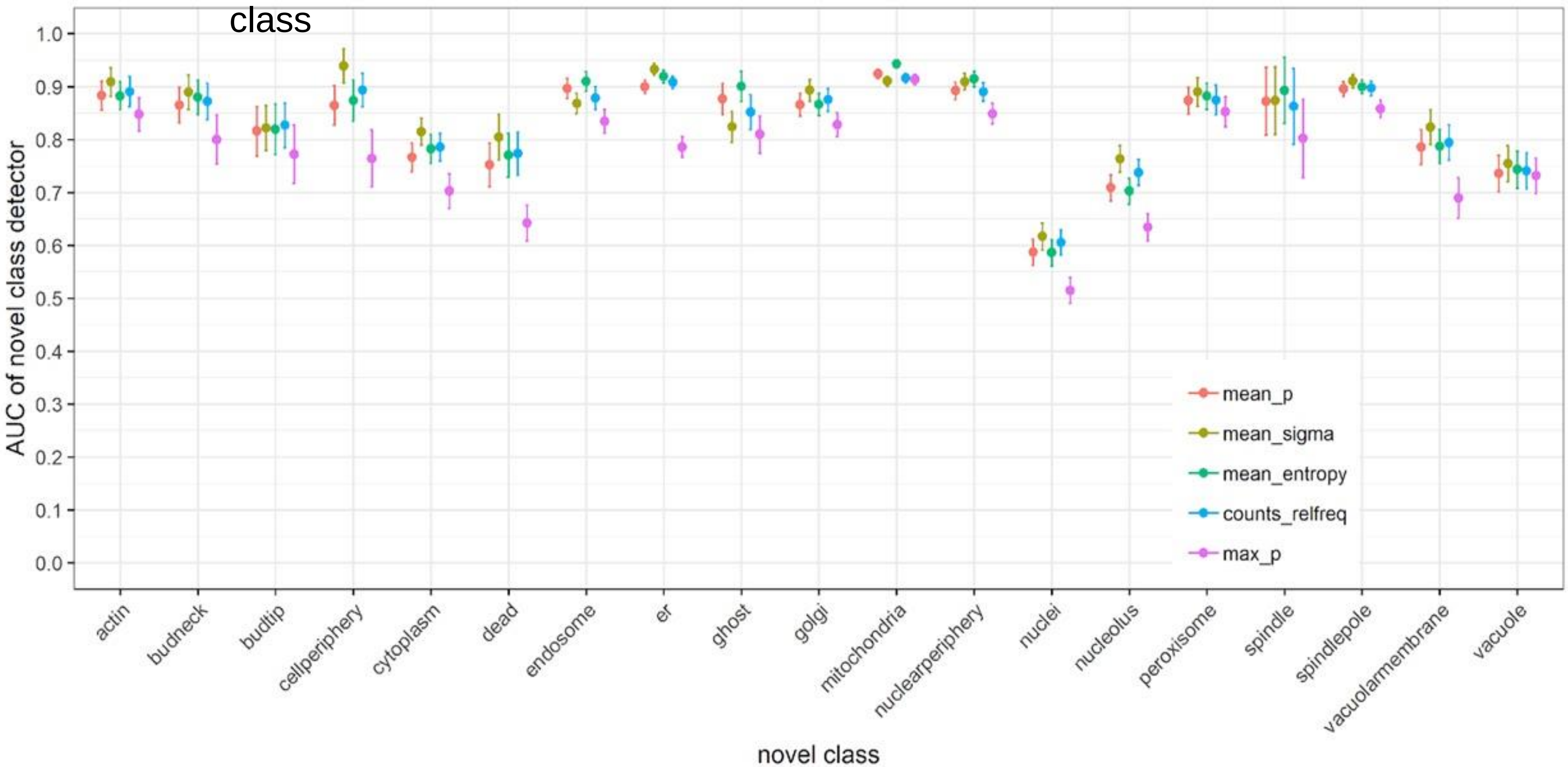
How good are novel/unusual classes identifiable?



All **MC Dropout based appoaches are superior compared** to the non-MC approach.

Dropout uncertainty measures outperform traditional CNN probability

CV experiment where each phenotyp is once in the role of the novel class



Creating custom layers in keras

```
from keras.layers.core import Lambda # needed to build the custom layer
from keras import backend as K #Now we have access to the backend (could be tensorflow,
```

Define your custom function

```
def mcdropout(x):
  #return tf.nn.dropout(x=x, keep_prob=0.33333) #using TensorFlow
  return K.dropout(x, level=0.5) # beeing agnostic of the backend
```

Here you could do anything possible in TF
`tf.add(x, 10)`
…

Include your custom function as a layer

```
model = Sequential()
model.add(Lambda(mcdropout, input_shape=(5,)))
#model.add(Dense(10))
#... Usually you would have many more layers
model.compile(loss='categorical_crossentropy',optimizer='adam')
```

https://github.com/oduerr/dl_tutorial/blob/master/tensorflow/keras/using_tf_in_keras.ipynb

Conclusion

MC Dropout during test time

→ yields uncertainty measures for each individual classification

→ helps to identify uncertain cases

→ allows to indicate novel classes

→ yields new probability estimates leading to higher accuracy