[Monte Carlo Dropout]

- Dropout : Switching-off some neurons at each training.
  At each step, a different set of neurons are switched off
  with some probability p (= dropout rate)
  0 ~ 0.5
  (no dropout) (50% dropout)

  prevent overfitting

- Why Dropout works :

  ① Information spreads out more evenly across the network. Hence, our neuron cannot rely on one or two inputs only, it has to spread out its weights and pay attention to all inputs.
  → less sensitive to input changes which results in the model generalizing better

  ② Since in every training iteration, you randomly sample the neurons to be dropped out in each layer (according to that layer's dropout rate), a different set of neurons are being dropped out each time.
  → Each time the model's architecture is slightly different
  → Outcome = averaging ensemble of many different neural networks, each trained on one batch of data only.

** dropout is used only during training.

Monte Carlo : a class of computational algorithms that rely on repeated random sampling to obtain a distribution of Some numerical quantity.

Regular Dropout = a Bayesian approximation of a well-known probabilistic model : Gaussian process.

How:

1. Add dropout at test time
2. average many predictions

softmax_output: `[0. 0. 1. 0. 0. 0. 0. 0. 0. 0.]`

mc_pred_proba: `[0. 0. 0.989 0.008 0.001 0. 0. 0.001 0.001 0. ]`

→ Better analysis on model's uncertainty

Conclusion

- Monte Carlo Dropout boils down to training a neural network with the regular dropout and keeping it switched on at inference time. This way, we can generate multiple different predictions for each instance.
- For classification tasks, we can average the softmax outputs for each class. This tends to lead to more accurate predictions, which additionally express the model's uncertainty properly.
- For regression tasks, we can analyze the predictive distribution to check which values are likely or summarize it using its mean or median.