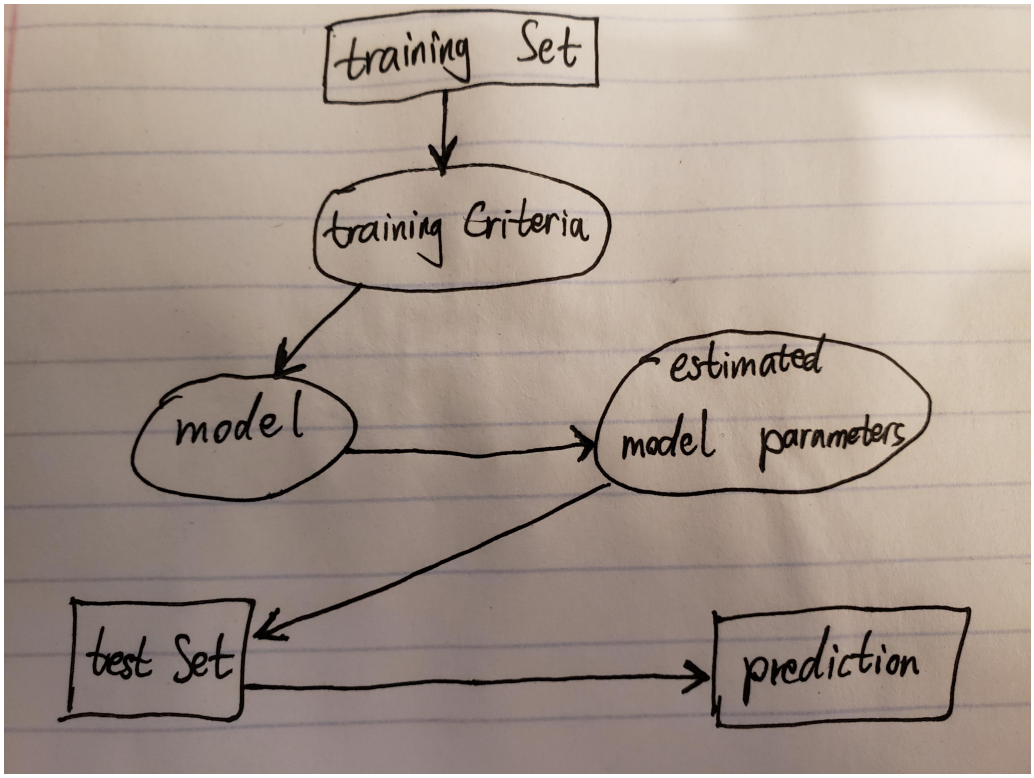
1. (1)



(2)

(a) Neural network: Model, estimated model parameters

(b) Smoothing using linear interpolation: Training and test sets

(c) p(e)=0.5: training criterion

(d) maximum likelihood: estimated model parameters

(e) mean square error: estimated model parameters

(f) the cat sat on the matt: Training Set, or Test Set

(g) matt the on: Prediction

2.

(1) **false**

(2) **False**

(3） **True**

(4) **A** and **C**

3.

(1) **D**

(2) **B** and **C**

4.

(1) auto-encoder is an **unsupervised learning** method.

Because it doesn’t use labeled data. We don’t provide any target and it will sets the target as the input.

It’s goal is to build a function to let the output mimic input.

(2) GMM is an **unsupervised learning** method.

It is a clustering model. Mixture models don't require knowing which class a data point belongs to, allowing the model to learn the class automatically. Since class assignment is not known, this constitutes a form of unsupervised learning.

5.

(1)

Neural network models are nonlinear and has is high flexible. The downside of flexible is **high variance** and it is not reliable to count on a single model to make decisions.

Training multiple models and combine the predictions can **reduce the variance and reduce generalization error**. This often has improved performance over any single network.

(2)

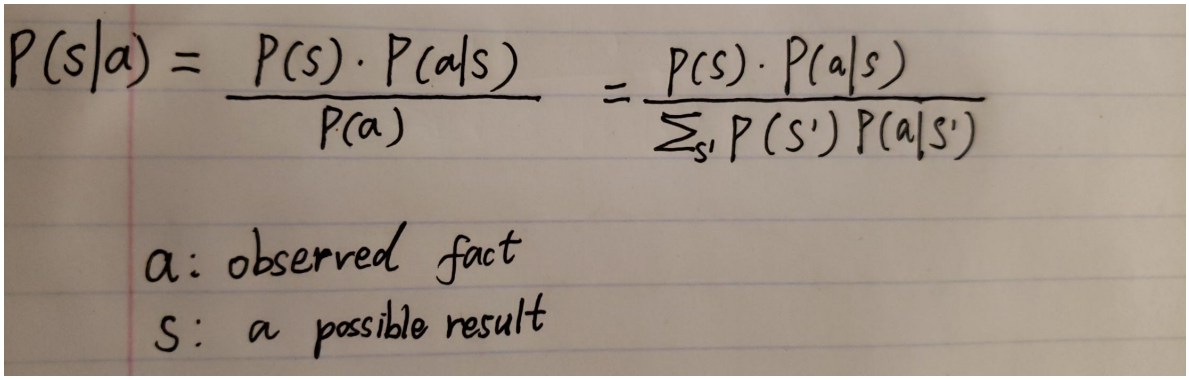
(a) A group of 10 models is almost always **less complex** than a single network that would achieve the same level of performance

(b) The risk of overfitting is eliminated, since there are **fewer parameters** needed in each model.

(c) A group of 10 models can be trained **more easily on smaller input sets**.

6.

(1)



(2)

Prior distribution: **p(s)**

likelihood term: **P(a|s)**

posterior distribution: **P(s|a)**

(3)

(a) MAP takes into account the prior probability of the considered hypotheses. ML does not.

(b) The MAP estimation procedure allows us to inject our prior beliefs about parameter values into the new estimate.

7.

(1) output_h

= sigmoid(w1 * x + bias1)

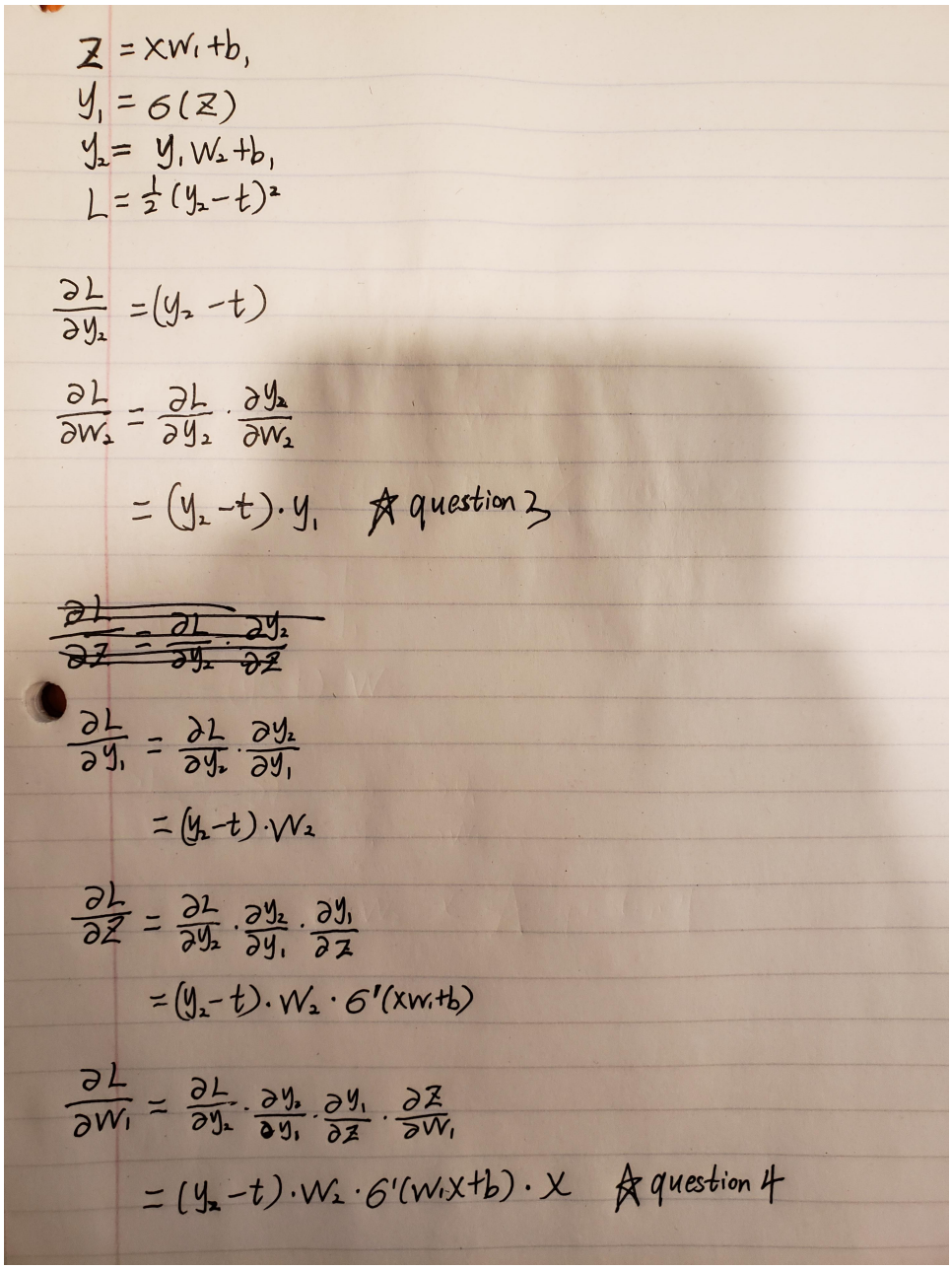
= sigmoid((-2) * 1 + (+2))= **0.5**

(2) y = w2 * output_h + bias2 = (+4) * 0.5 + (0) = 2

Loss = (target-y)^2 / 2

= (1 - 2)^2 / 2 = **0.5**

(3) and (4)



8. The problem is that the **order** of a word in a sentence **matters**. If we use shared weight, then **the information about words' order were lost**. It makes no difference to be the first word or be the second word.

This is called rank disorder, it will cause them to ignore networks that achieve high accuracy when their parameters are trained without sharing.

9.

(1) **Bias decrease, Variance increase.**

Because adding more hidden unites **makes the model more complicated**. And in general, more complicated models have lower bias but higher variance

(2) If tuning hyperparameters using test set, then we are actually selecting hyperparameters that work well with the set.

But the performance can't be measured on unseen data. In simple words, **generalization performance isn't be measured**. Because the parameters are already tuned specifically for test set.

10.

(1) Solution set: **w1 * w2 = 1.**

Prove:

L = 1/2 (x - w2 * w1 * x)^2 = 0

x - w2 * w1 * x = 0

x = w2 * w1 * x

w1*w2 = 1

(2) **Yes,** it has a saddle point.

The saddle point is **w1 = w2 = 0**.