CSC321: Introduction to Neural Networks and Machine Learning

Lecture 15: Mixtures of Experts

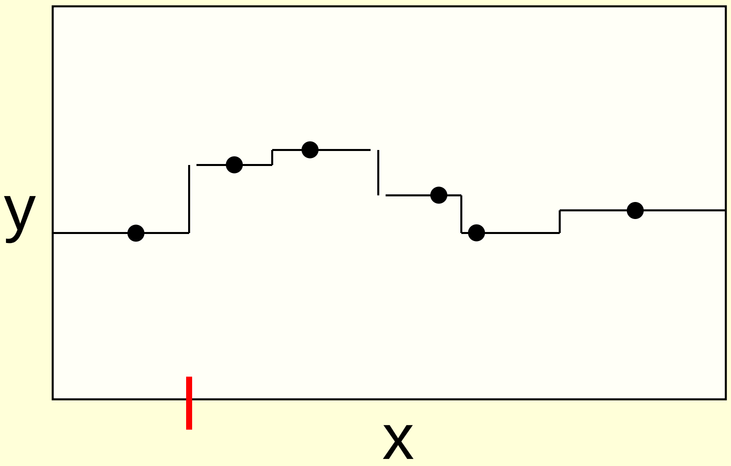
Geoffrey Hinton

A spectrum of models

Very local models

- e.g. Nearest neighbors

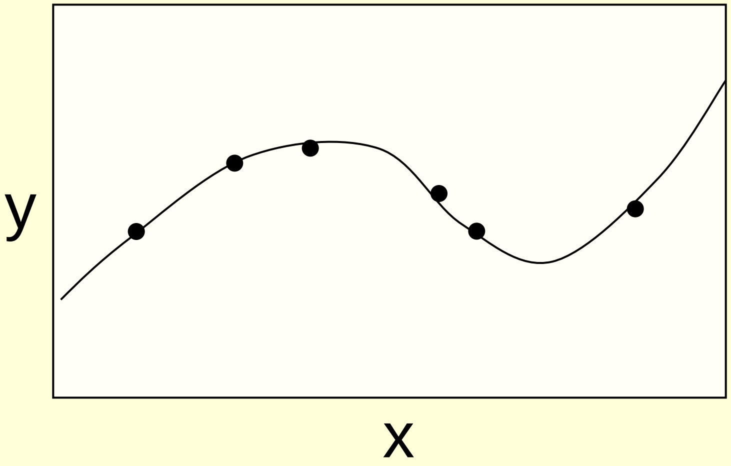
- Very fast to fit
  - Just store training cases
- Local smoothing obviously improves things



Fully global models

- e. g. Polynomial

- May be slow to fit
  - Each parameter depends on all the data

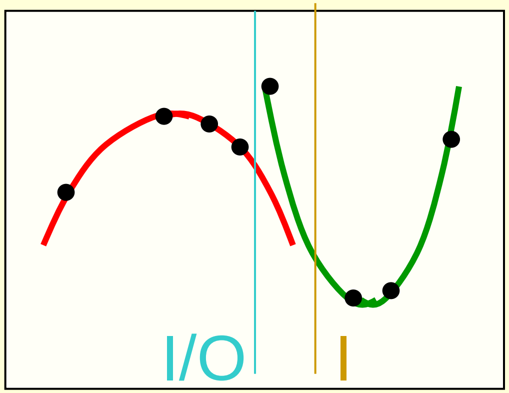


Multiple local models

- Instead of using a single global model or lots of very local models, use several models of intermediate complexity.
  - Good if the dataset contains several different regimes which have different relationships between input and output.
  - But how do we partition the dataset into subsets for each expert?

Partitioning based on input alone versus partitioning based on input-output relationship

- We need to cluster the training cases into subsets, one for each local model.
  - The aim of the clustering is NOT to find clusters of similar input vectors.
  - We want each cluster to have a relationship between input and output that can be well-modeled by one local model

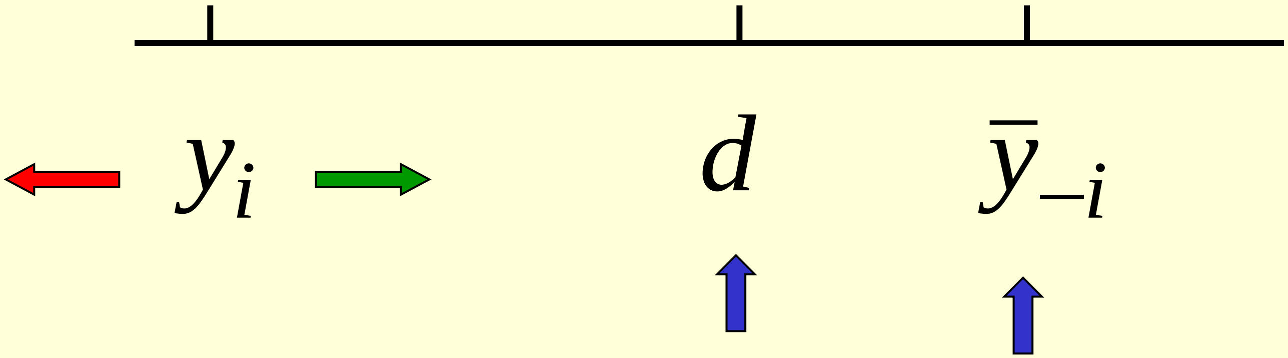


which partition is best:
I=input alone or I/O=input→output mapping?

Mixtures of Experts

- Can we do better that just averaging predictors in a way that does not depend on the particular training case?
  - Maybe we can look at the input data for a particular case to help us decide which model to rely on.
    - This may allow particular models to specialize in a subset of the training cases. They do not learn on cases for which they are not picked. So they can ignore stuff they are not good at modeling.
- The key idea is to make each expert focus on predicting the right answer for the cases where it is already doing better than the other experts.
  - This causes specialization.
  - If we always average all the predictors, each model is trying to compensate for the combined error made by all the other models.

A picture of why averaging is bad



target

Average of all the other predictors

Do we really want to move the output of predictor i away from the target value?

Making an error function that encourages specialization instead of cooperation

- If we want to encourage cooperation, we compare the average of all the predictors with the target and train to reduce the discrepancy.
  - This can overfit badly. It makes the model much more powerful than training each predictor separately.
- If we want to encourage specialization we compare each predictor separately with the target and train to reduce the average of all these discrepancies.
  - Its best to use a weighted average, where the weights, p, are the probabilities of picking that "expert" for the particular training case.

Average of all the predictors

$$E = (d - \langle y_i \rangle_i)^2$$

$$E = \Big\langle p_i (d - y_i)^2 \Big\rangle_i$$

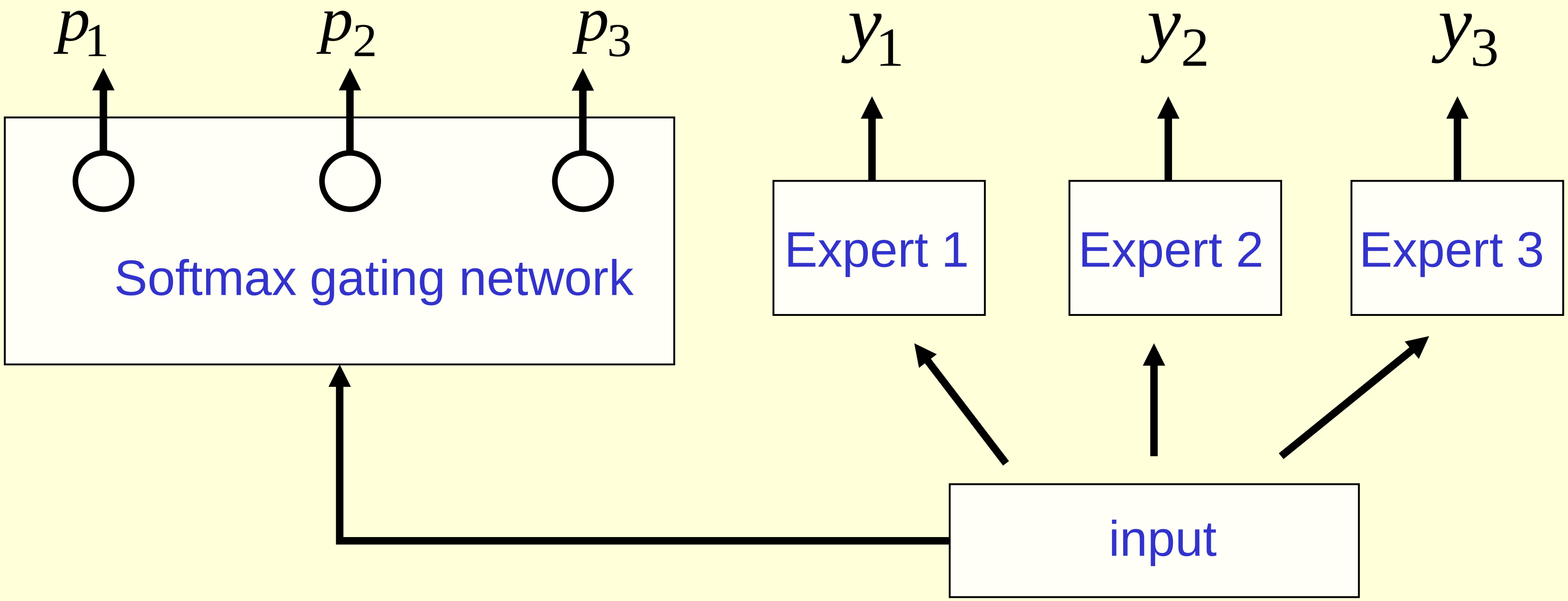
probability of picking expert i for this case

The mixture of experts architecture

Combined predictor: $$y = \sum_i p_i \, y_i$$

Simple error function for training: $$E = \sum_i p_i (d - y_i)^2$$

(There is a better error function)



The derivatives of the simple cost function

- If we differentiate w.r.t. the outputs of the experts we get a signal for training each expert.
- If we differentiate w.r.t. the outputs of the gating network we get a signal for training the gating net.
  - We want to raise p for all experts that give less than the average squared error of all the experts (weighted by p)

$$E = \sum_i p_i (d - y_i)^2$$

$$\frac{\partial E}{\partial y_i} = p_i (d - y_i)$$

$$\frac{\partial E}{\partial x_i} = p_i [(d - y_i)^2 - E]$$

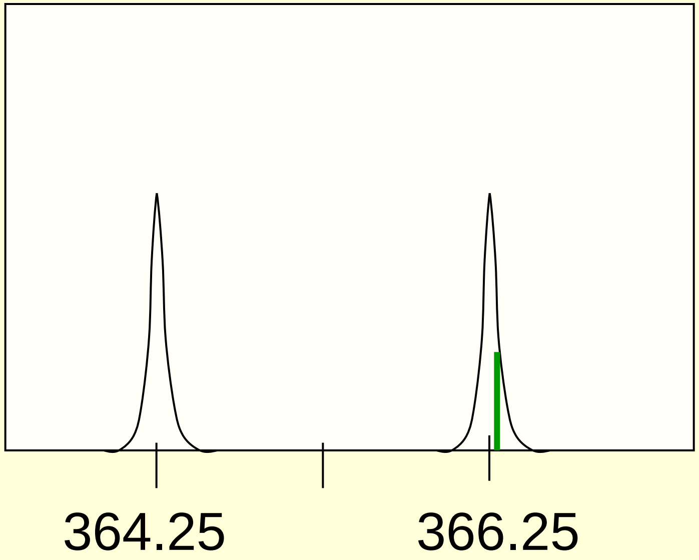
$$where \quad p_i = \frac{e^{x_i}}{\sum_j e^{x_j}}$$

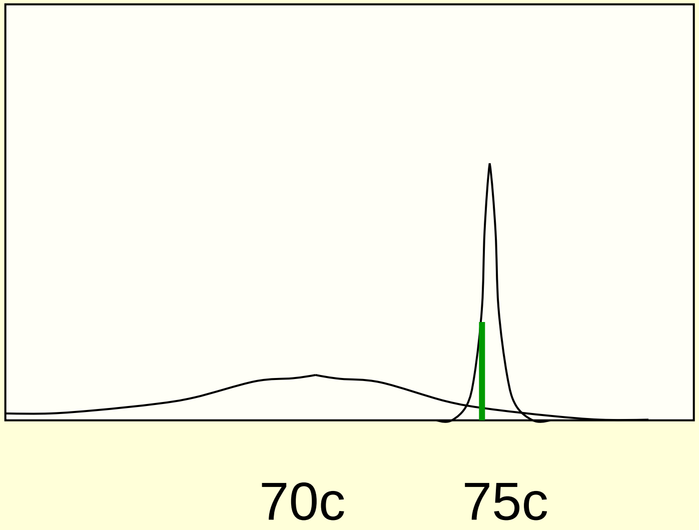
Another view of mixtures of experts

- One way to combine the outputs of the experts is to take a weighted average, using the gating network to decide how much weight to place on each expert.
- But there is another way to combine the experts.
  - How many times does the earth rotate around its axis each year?
  - What will be the exchange rate of the Canadian dollar the day after the Quebec referendum?

Giving a whole distribution as output

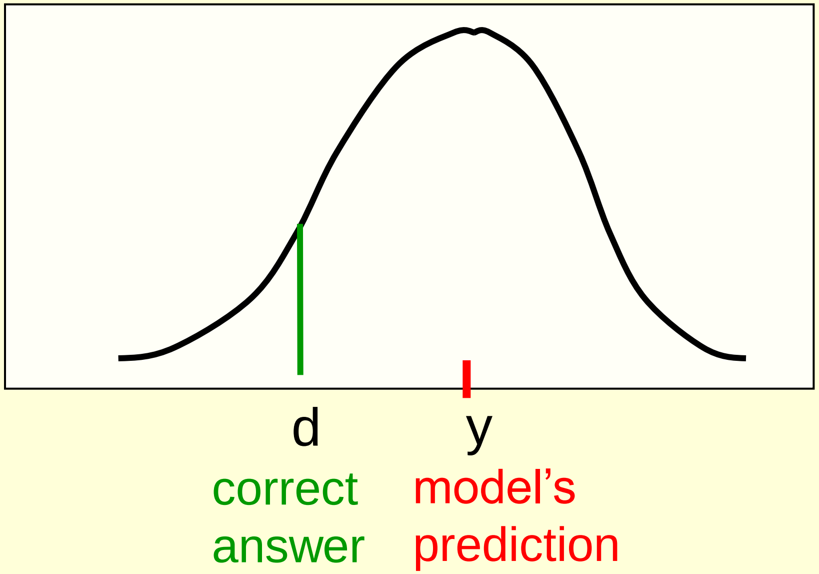
- If there are several possible regimes and we are not sure which one we are in, its better to output a whole distribution.
  - – Error is negative log probability of right answer





The probability distribution that is implicitly assumed when using squared error

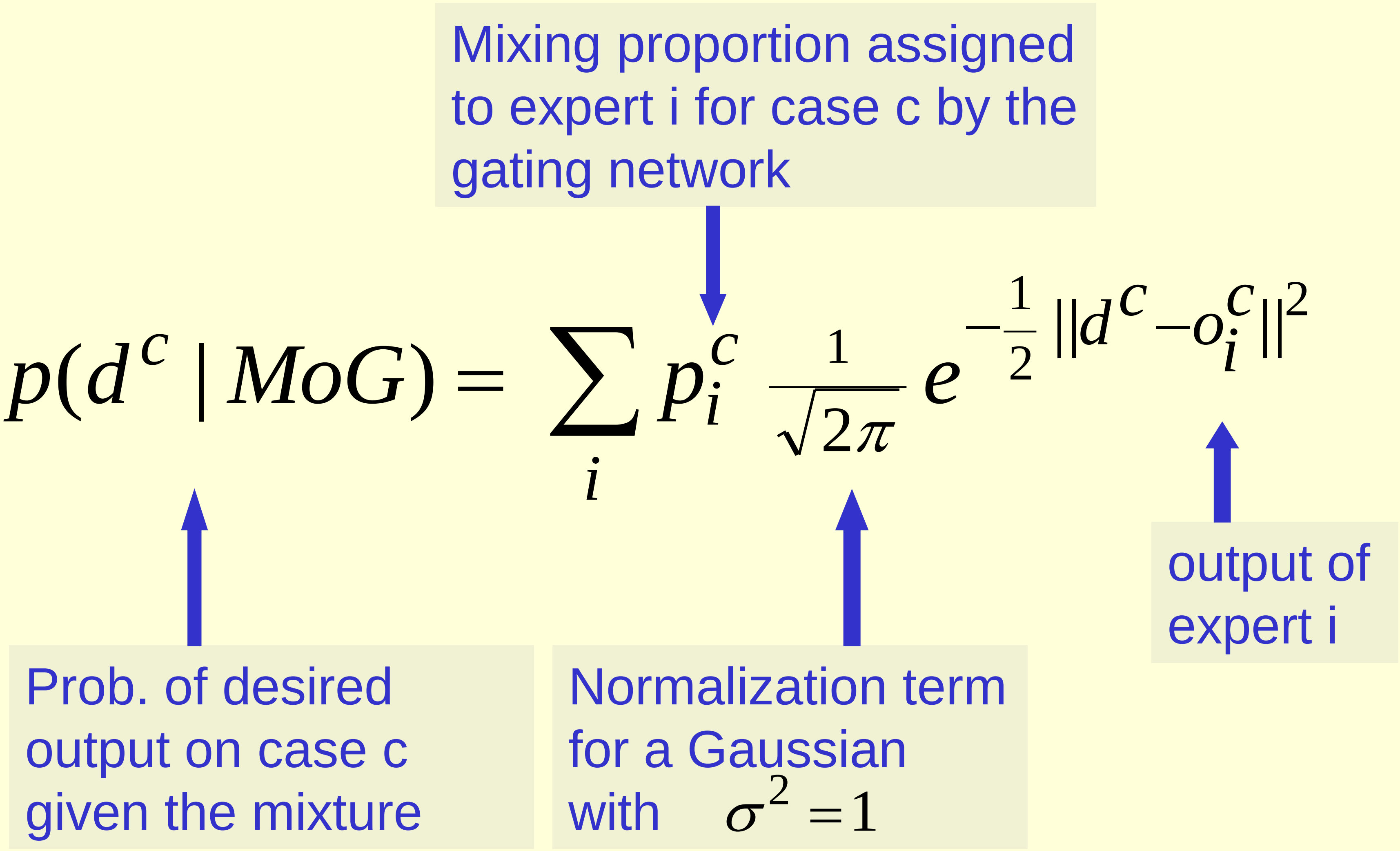
- Minimizing the squared residuals is equivalent to maximizing the log probability of the correct answers under a Gaussian centered at the model's guess.
  - – If we assume that the variance of the Gaussian is the same for all cases, its value does not matter.



$$p(d) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(d-y)^2}{2\sigma^2}}$$

$$-\log p(d) = k + \frac{(d-y)^2}{2\sigma^2}$$

The probability of the correct answer under a mixture of Gaussians



Mixing proportion assigned to expert i for case c by the gating network

$$p(d^c \mid MoG) = \sum_i p_i^c \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}||d^c - o_i^c||^2}$$

Prob. of desired output on case c given the mixture

Normalization term for a Gaussian with $\sigma^2 = 1$

output of expert i

A natural error measure for a Mixture of Experts

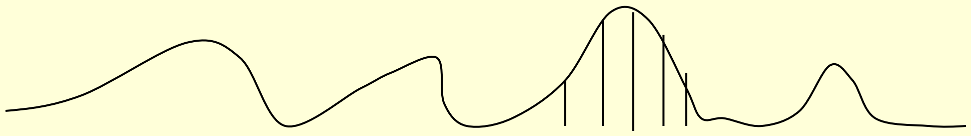
$$-\log p(d^c \mid MoE) = -\log \sum_i p_i^c \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}||d^c - o_i^c||^2}$$

$$\frac{\partial E^c}{\partial o_i^c} = -2 \boxed{\frac{p_i^c e^{-\frac{1}{2}||d^c - o_i^c||^2}}{\sum_j p_j^c e^{-\frac{1}{2}||d^c - o_j^c||^2}}} (d^c - o_i^c)$$

This fraction is the posterior probability of expert i

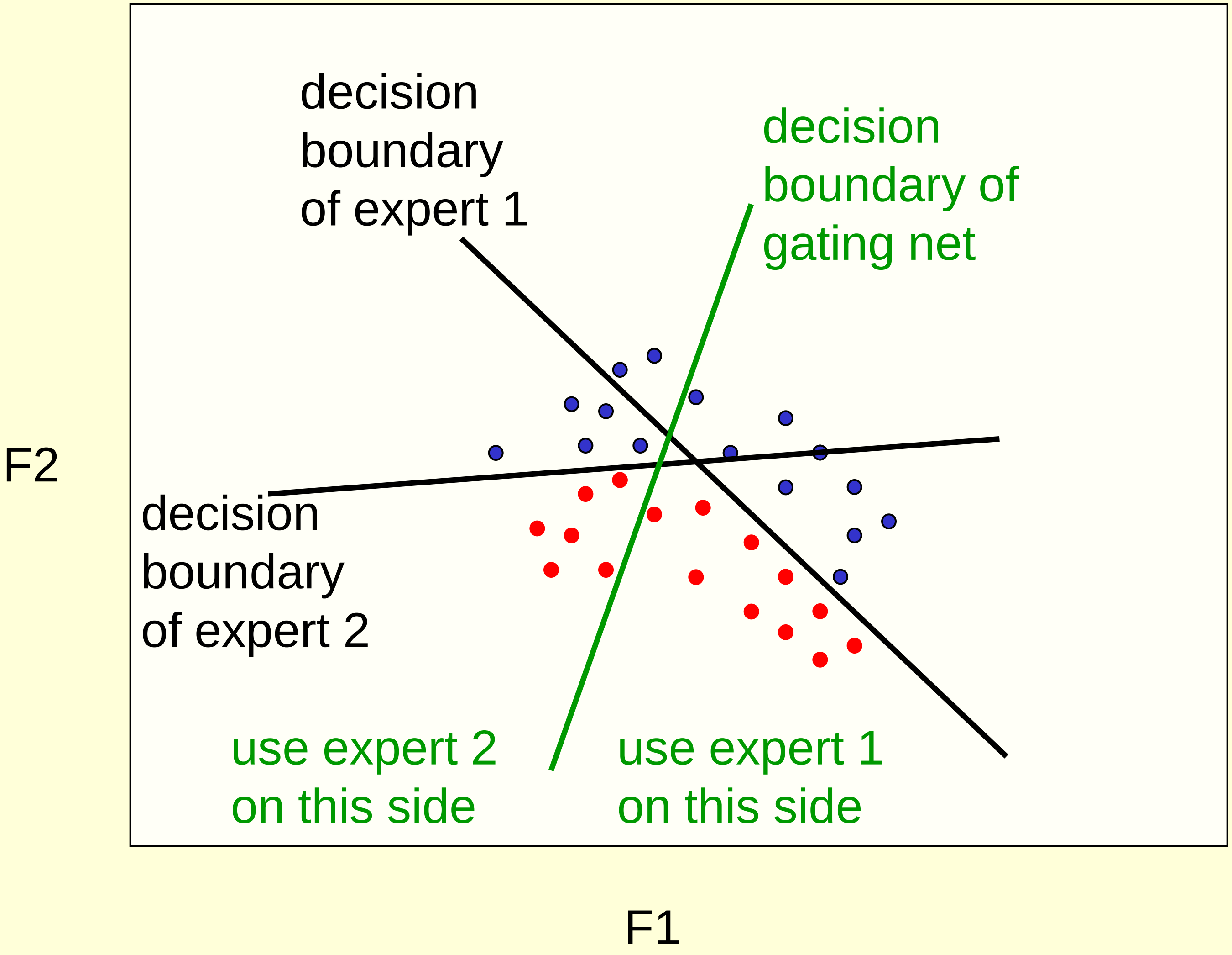
What are vowels?

- The vocal tract has about four resonant frequencies which are called formants.
  - We can vary the frequencies of the four formants.
- How do we hear the formants?
  - The larynx makes clicks. We hear the dying resonances of each click.
  - The click rate is the pitch of the voice. It is independent of the formants. The relative energies in each harmonic of the pitch define the envelope of a formant.



- Each vowel corresponds to a different region in the plane defined by the first two formants, F1 and F2. Diphthongs are different.

A picture of two imaginary vowels and a mixture of two linear experts after learning



decision boundary of expert 1
decision boundary of gating net
F2
decision boundary of expert 2
use expert 2 on this side
use expert 1 on this side
F1

Decision trees

- In a decision tree, we start at the root node and perform a test on the input vector to determine whether to take the right or left branch.
- At each internal node of the tree we have a different test, and the test is usually some very simple function of the input vector.
  - Typical test: Is the third component of the input vector bigger than 0.7?
- One we reach a leaf node, we apply a function to the input vector to compute the output.
  - The function is specific to that leaf node, so the sequence of tests is used to pick an appropriate function to apply to the current input vector.

Decision Stumps

- Consider a decision tree with one root node directly connected to N different leaf nodes.
  - The test needs to have N possible outcomes.
- Each leaf node is an “expert” that uses its own particular function to predict the output from the input.
- Learning a decision stump is tricky if the test has discrete outcomes because we do not have a continuous space in which to optimize parameters.

Creating a continuous search space for decision stumps

- If the test at the root node uses a softmax to assign probabilities to the leaf nodes we get a continuous search space:
  - – Small changes to the parameters of the softmax “manager” cause small changes to the expected log probability of predicting the correct answer.
- A mixture of experts can be viewed as a probabilistic way of viewing a decision stump so that the tests and leaf functions can be learned by maximum likelihood.
  - – It can be generalised to a full decision tree by having a softmax at each internal node of the tree.