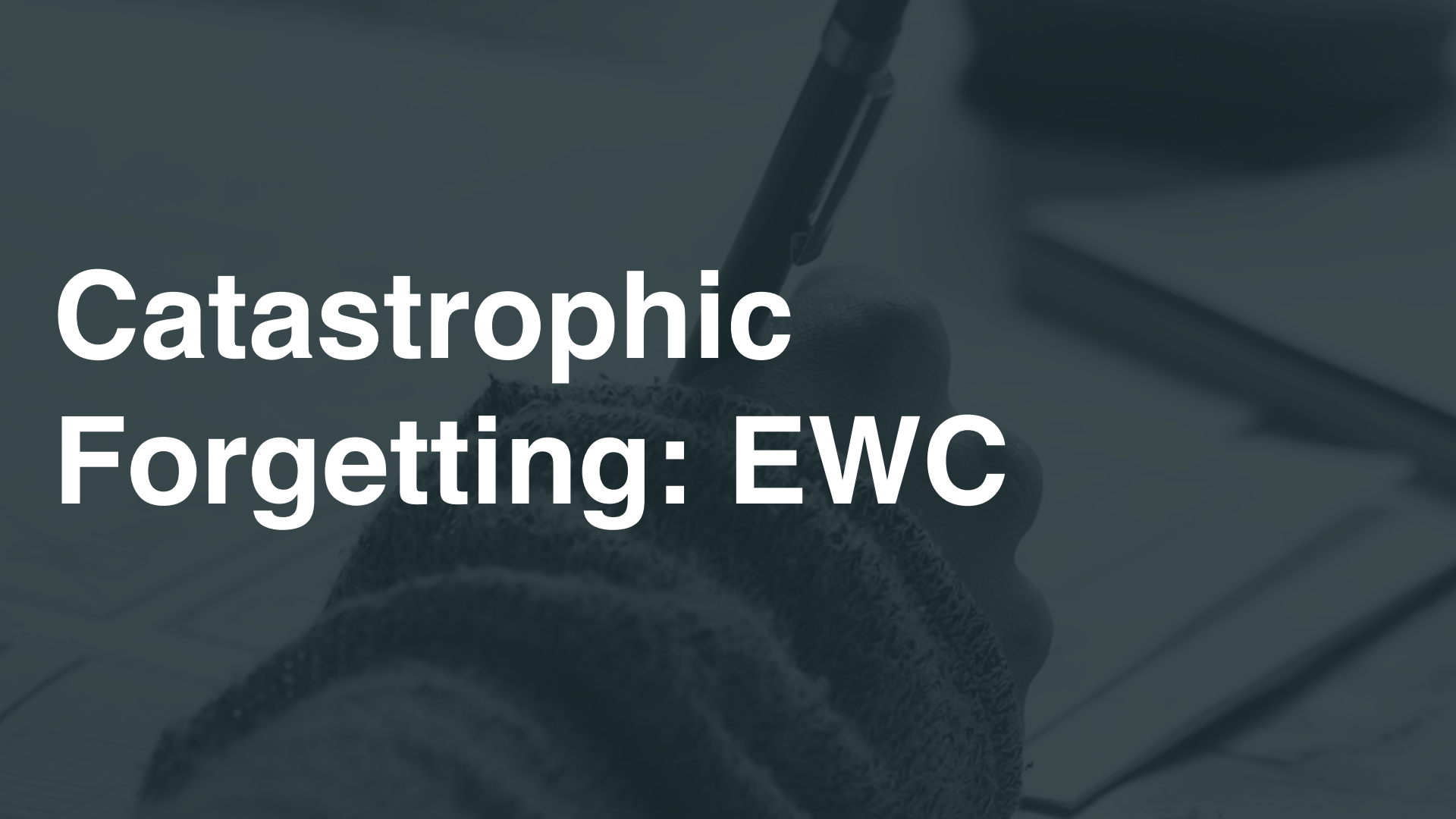
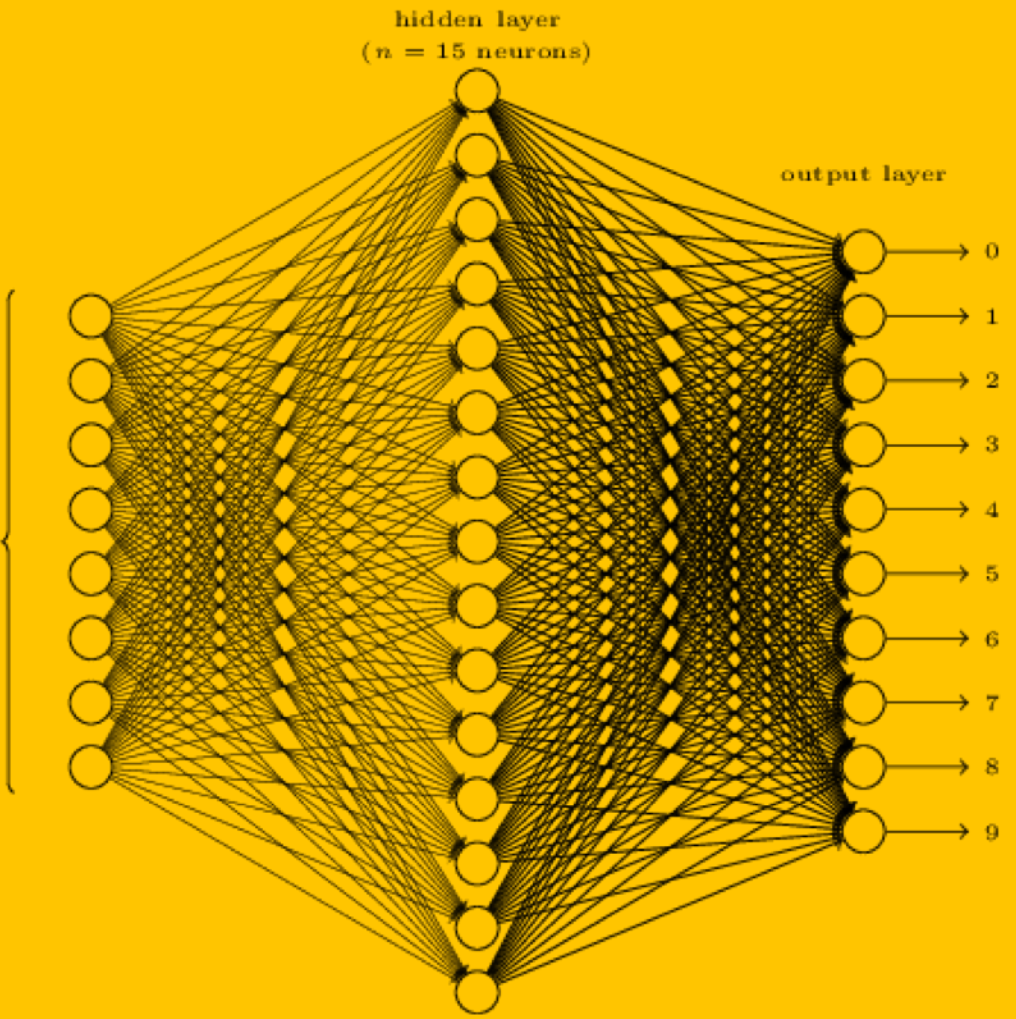


Catastrophic Forgetting: EWC


hidden layer
(n = 15 neurons)
output layer
input layer
(784 neurons)
0
1
2
3
4
5
6
7
8
9

McCloskey and Cohen (1989)

17 single-digit ones problems (i.e., 1 + 1 through 9 + 1, and 1 + 2 through 1 + 9)
then
17 single-digit twos problems (i.e., 2 + 1 through 2 + 9, and 1 + 2 through 9 + 2)

- interference will occur when new learning alters the weights involved representing something
- the greater the amount of new learning, the greater the disruption in old knowledge
- interference was catastrophic in the backpropagation networks when learning was sequential but not concurrent

Solution Attempts

Activation Sharpening (Sharpen the activation of most active nodes) | French (1991)

Pre-training (Train net on similar data) | McRae and Hetherington (1993)

Orthogonal representations (sum product of pairwise patterns to zero) | Lewandowsky and Li (1995)

*Complementary Learning System*s (slow learning/fast learning) | McClelland (1995)

Self refreshing memory (Interleave new representations with past ones) | Ans and Rousset (1997)

Synaptic Consolidation

connections between neurons are less likely to be overwritten if they have been important in previously learnt tasks

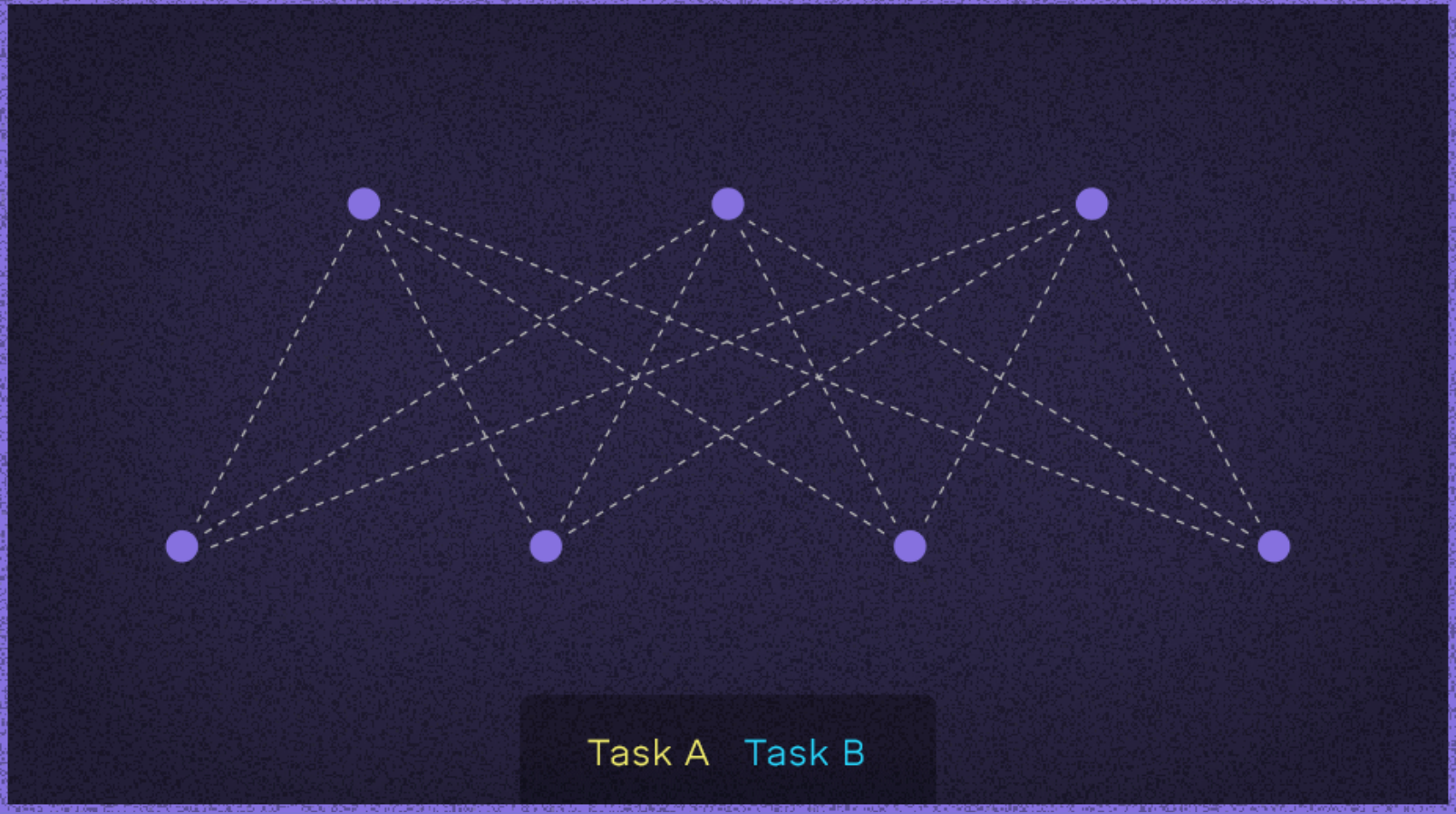
Heard of synaptic plasticity? Long term potentiation?

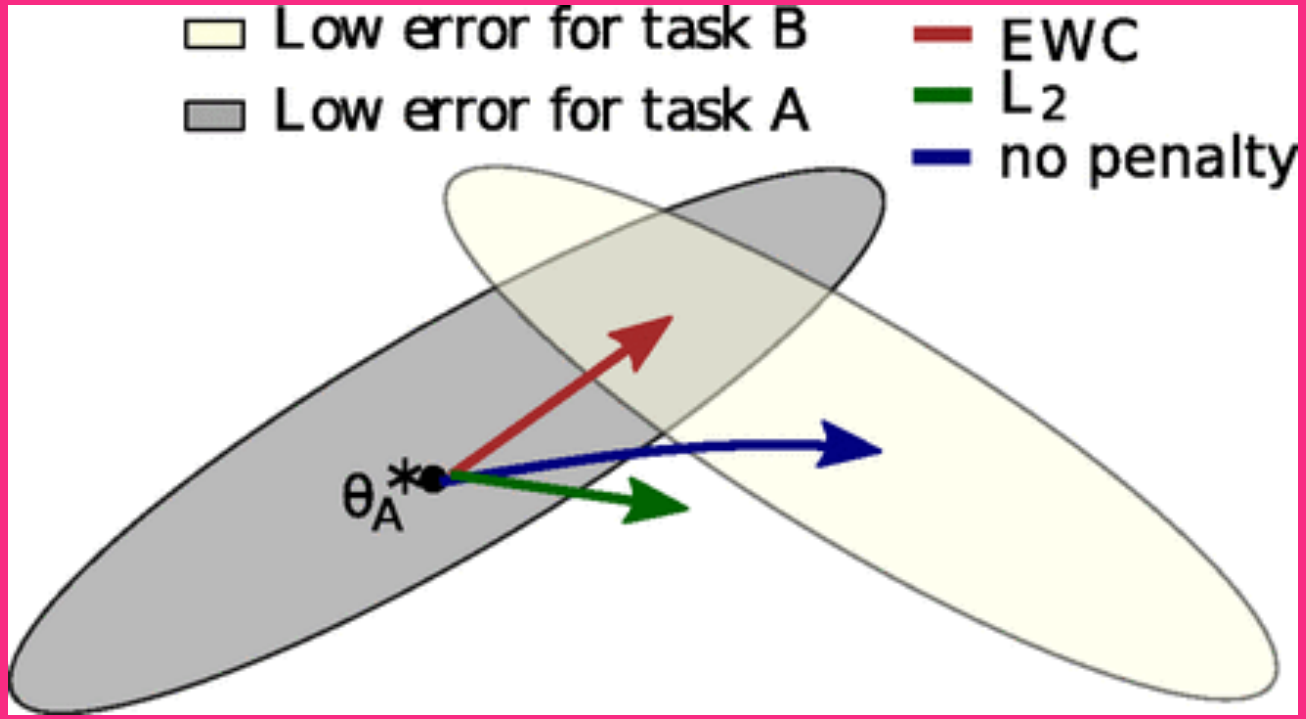
Thats *small* (Clopath 2016)

Elastic Weight Consolidation

In a nutshell: Weight *plasticity changes as a function of task importance

*or… *elasticity*


Task A
Task B



Weight parameter θ_* alterations that optimize remembrance for previously learned tasks

(Takes advantage of the possible weight combinatorics of DNN)

To justify this choice of constraint and to define which weights are most important for a task, it is useful to consider neural network training from a probabilistic perspective. From this point of view, optimizing the parameters is tantamount to finding their most probable values given some data $\mathscr{D}$. We can compute this conditional probability $p\left(\theta|\mathscr{D}\right)$ from the prior probability of the parameters $p\left(\theta\right)$ and the probability of the data $p\left(\mathscr{D}|\theta\right)$ by using Bayes' rule:

$$\log p\left(\theta|\mathscr{D}\right) = \log p\left(\mathscr{D}|\theta\right) + \log p\left(\theta\right) - \log p\left(\mathscr{D}\right).$$

AKA: Bayesians Don't forget: full Bayesian posterior distribution of weights that worked.

Use this as a prior for the K+1 task, then update the posterior with the new info. Rinse, repeat.

To justify this choice of constraint and to define which weights are most important for a task, it is useful to consider neural network training from a probabilistic perspective. From this point of view, optimizing the parameters is tantamount to finding their most probable values given some data $\mathscr{D}$. We can compute this conditional probability $p(\theta|\mathscr{D})$ from the prior probability of the parameters $p(\theta)$ and the probability of the data $p(\mathscr{D}|\theta)$ by using Bayes' rule:

$$\log p(\theta|\mathscr{D}) = \log p(\mathscr{D}|\theta) + \log p(\theta) - \log p(\mathscr{D}).$$

~~AKA: Bayesians Don't forget: full Bayesian posterior distribution of weights that worked.~~

~~Use this as a prior for the K+1 task, then update the posterior with the new info. Rinse, repeat.~~

This is intractable

How to bridge the gap between the statistical awesomeness of Bayesian inference and the efficiency of gradient descent?

Most important weights?

$$\mathcal{L}(\theta) = \mathcal{L}_B(\theta) + \sum_i \frac{\lambda}{2} F_i (\theta_i - \theta^*_{A,i})^2,$$

Loss function to minimize

Most important weights?

$$\mathscr{L}(\theta) = \mathscr{L}_B(\theta) + \sum_i \frac{\lambda}{2} F_i (\theta_i - \theta^*_{A,i})^2,$$

on-line diagonalised Laplace approximation approach

similar to assumed density filtering (ADF, Opper & Winther, 1999)

precursor to expectation-propagation (Minka, 2001)

Most important weights?

$$\mathscr{L}(\theta) = \mathscr{L}_B(\theta) + \sum_i \frac{\lambda}{2} F_i (\theta_i - \theta^*_{A,i})^2,$$

on-line diagonalised Laplace approximation approach

similar to assumed density filtering (ADF, Opper & Winther, 1999)

precursor to expectation-propagation (Minka, 2001)

Most important weights?

$$\mathscr{L}(\theta) = \mathscr{L}_B(\theta) + \sum_i \frac{\lambda}{2} F_i (\theta_i - \theta^*_{A,i})^2,$$

Approximate the probability distribution with a Gaussian

- mean is at the mode of the distribution
- variance is given by the inverse diagonal entries of the Hessian of the log density at the mode
  - (Fisher information)

Same for the third task?

There is a bit of controversy…. Also reddit.

www.inference.vc/comment-on-overcoming-catastrophic-forgetting-in-nns-are-multiple-penalties-needed-2/

Do we actually need to remember θ^A? $\theta^{A,B}$ is supposed to be the mode of the posterior $p(\theta|\mathcal{D}_A, \mathcal{D}_B)$ and the posterior already captures all our knowledge about *both tasks* A and B. The mode of the previous posterior θ^A should become irrelevant as it is already incorporated into $\theta^{A,B}$. Somehow, this just doesn't feel right.

Let's apply the sequential diagonal Laplace approximation argument consistently, but now for learning the third task C, after A and B. For this, we need the mode and Hessian of the posterior $p(\theta|\mathcal{D}_A, \mathcal{D}_B, \mathcal{D}_C)$, which can be expressed as:

$$\log p(\theta|\mathcal{D}_A, \mathcal{D}_B, \mathcal{D}_C) = -\mathcal{L}_C(\theta) + \log p(\theta|\mathcal{D}_A, \mathcal{D}_B) + \text{constant}$$

Let's replace the intractable $\log p(\theta|\mathcal{D}_A, \mathcal{D}_B)$ with its Laplace approximation, once again. We have already calculated the mode of this distribution, it is (approximately) $\theta^{A,B}$. How about its Hessian around $\theta^{A,B}$? Well, we have assumed that

$$\log p(\theta|\mathcal{D}_A, \mathcal{D}_B) \approx -\mathcal{L}_B(\theta) - \sum_i F^A_{i,i}(\theta_i - \theta^A_i)^2 + \text{constant}$$

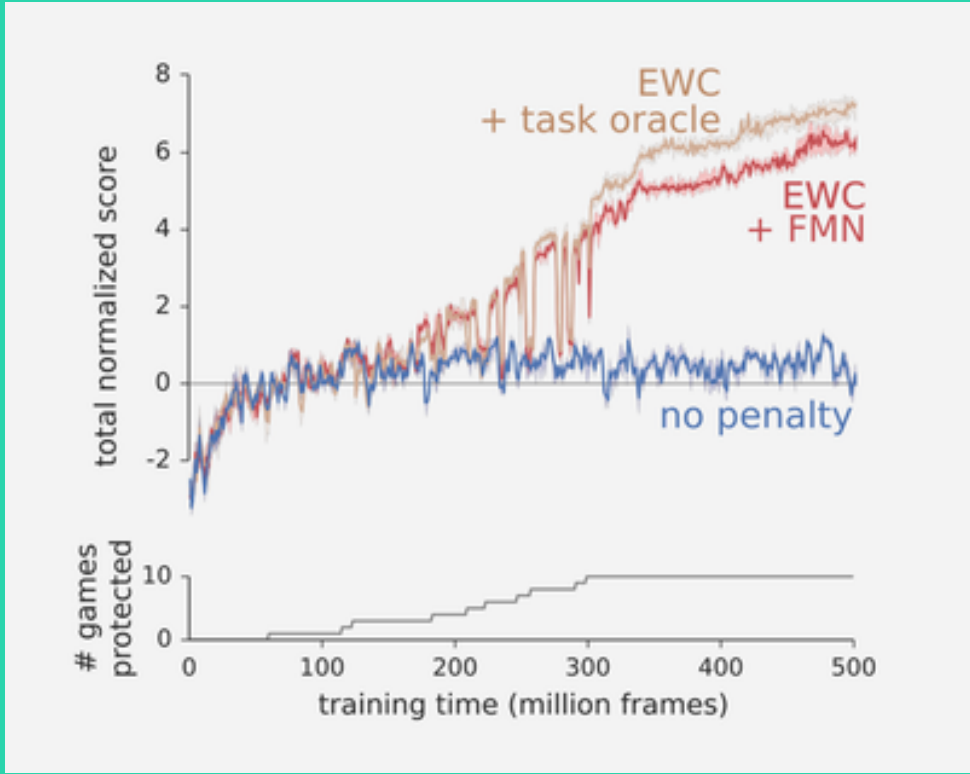
so the Hessian is the Fisher information matrix F^B plus the previously assumed diagonal approximation $diag(F^A)$. So, if we plug these in to form a diagonal Laplace approximation to $\log p(\theta|\mathcal{D}_A, \mathcal{D}_B)$, we get:

$$\log p(\theta|\mathcal{D}_A, \mathcal{D}_B) \approx -\sum_i (F^A + F^B)_{i,i}(\theta_i - \theta^{A,B}_i)^2 + \text{constant}$$

Games anyone?

Sequential exposure to atari games.

(average games learned)



synapse: the weight, its variance and its mean