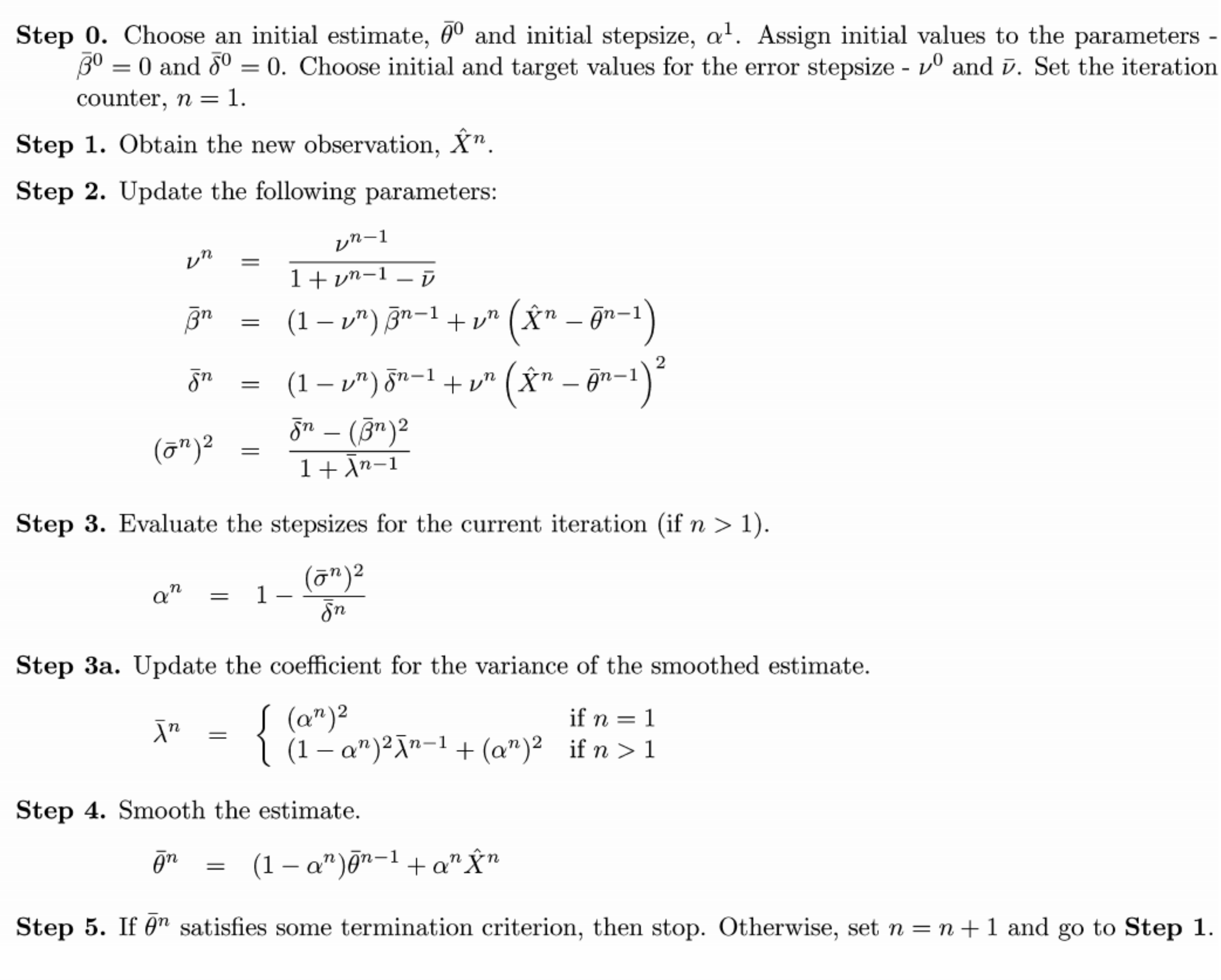
A Optimal Step-Size Adaptation

George and Powell [2006] propose Optimal Step-Size Adaptation (OSA), an algorithm for optimally estimating the value of a parameter θ given only noisy samples $x^n = \theta^n + \epsilon^n$ (n is a superscript indexing the ordered sequence of samples). The algorithm assumes that ϵ is some zero-mean IID noise and that we generate our estimates $\bar{\theta}^n$ by averaging: $\bar{\theta}^n = (1-\alpha^n)\bar{\theta}^{n-1} + \alpha^n x^n$. The algorithm is "optimal" in the sense that the step sizes α^n are optimal if the bias $\beta = \theta^n - \mathbb{E}[\bar{\theta}^{n-1}]$ in θ and the magnitude σ of the noise are known. Since they are generally not, OSA also estimates these from the input stream using running averages. The algorithm, copied directly from their paper, is as follows:



Step 0. Choose an initial estimate, $\bar{\theta}^0$ and initial stepsize, α^1. Assign initial values to the parameters - $\bar{\beta}^0 = 0$ and $\bar{\delta}^0 = 0$. Choose initial and target values for the error stepsize - ν^0 and $\bar{\nu}$. Set the iteration counter, $n = 1$.

Step 1. Obtain the new observation, $\hat{X}^n$.

Step 2. Update the following parameters:

$$\begin{aligned} \nu^n &= \frac{\nu^{n-1}}{1+\nu^{n-1}-\bar{\nu}} \\ \bar{\beta}^n &= (1-\nu^n)\,\bar{\beta}^{n-1} + \nu^n\left(\hat{X}^n - \bar{\theta}^{n-1}\right) \\ \bar{\delta}^n &= (1-\nu^n)\,\bar{\delta}^{n-1} + \nu^n\left(\hat{X}^n - \bar{\theta}^{n-1}\right)^2 \\ (\bar{\sigma}^n)^2 &= \frac{\bar{\delta}^n - (\bar{\beta}^n)^2}{1+\bar{\lambda}^{n-1}} \end{aligned}$$

Step 3. Evaluate the stepsizes for the current iteration (if $n > 1$).

$$\alpha^n = 1 - \frac{(\bar{\sigma}^n)^2}{\bar{\delta}^n}$$

Step 3a. Update the coefficient for the variance of the smoothed estimate.

$$\bar{\lambda}^n = \begin{cases} (\alpha^n)^2 & \text{if } n = 1 \\ (1-\alpha^n)^2\bar{\lambda}^{n-1} + (\alpha^n)^2 & \text{if } n > 1 \end{cases}$$

Step 4. Smooth the estimate.

$$\bar{\theta}^n = (1-\alpha^n)\bar{\theta}^{n-1} + \alpha^n\hat{X}^n$$

Step 5. If $\bar{\theta}^n$ satisfies some termination criterion, then stop. Otherwise, set $n = n + 1$ and go to **Step 1**.

B Hyperparameter Search

The parameters used in Figure 2 were obtained by running a hyperparameter search which looked for hyperparameters which led to the smallest average error at the end of each phase (i.e. at t=249 and t=499). The search used a Gaussian Process optimizer with 500 iterations. We found that there generally tends to be a large, flat region in parameter space with reasonably "good" parameters, as is evidence by the large regions of purple in Figure 4.

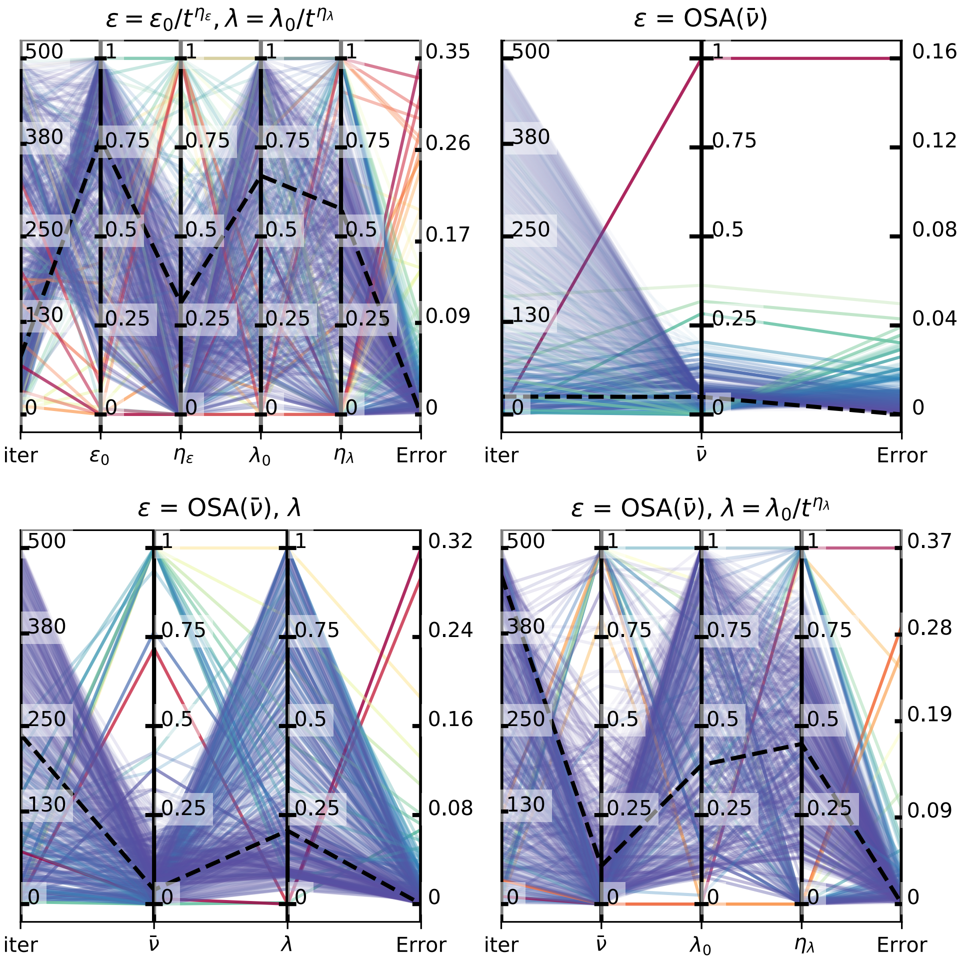


Figure 4: Parallel-Coordinates plot visualizing the hyperparameter search for the best-converging parameters. Each plot corresponds to one of the lines in Figure 2 (excluding the first, $1/\sqrt{t}$, which has no hyperparameters). In each plot, the leftmost axis represents the iteration in the Gaussian-Process search, the middle axes represent the hyperparameter values, the rightmost axis represents the error resulting from that set of hyperparameters (in this case the mean of the distance from the true fixed point at the end of each phase - see Figure 2), and lines are also colour-coded to indicate the error (purple is low and red is high). The black dotted line indicates the final selected set of hyperparameters. Large purple regions in these plots indicate insensitivity to hyperparameter values.

C Details on MNIST Experiments

For the MNIST experiments, we ran the hyperparameter optimization for both scheduled annealing and OSA + predictive coding. We optimized the hyperparameters based on the validation error after 1 epoch of training. Figure 5 visualizes the hyperparameter search. Because the two performed similarly and the latter had fewer parameters, we chose OSA + predictive coding for the experiment in Figure 3.

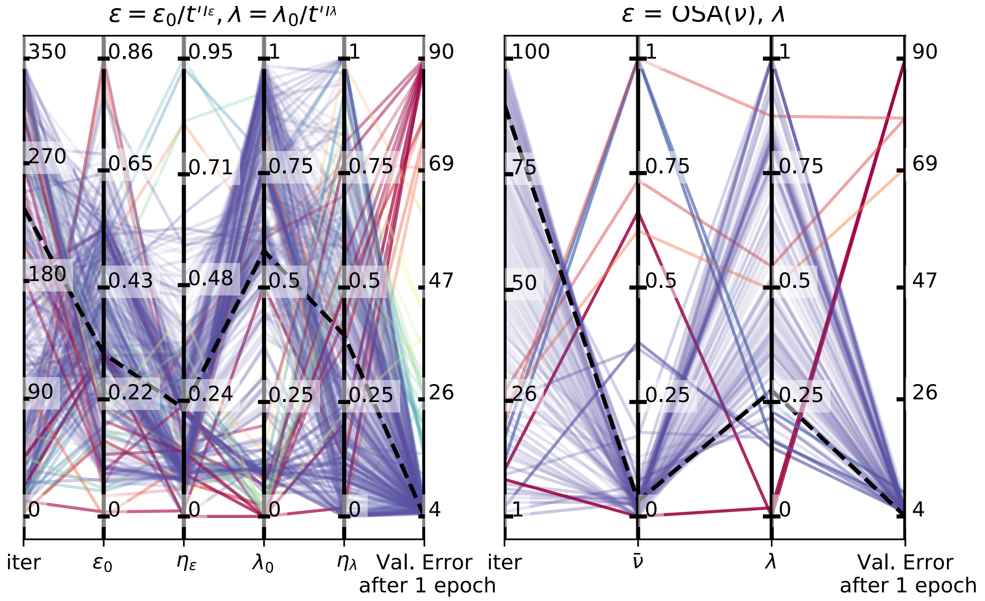
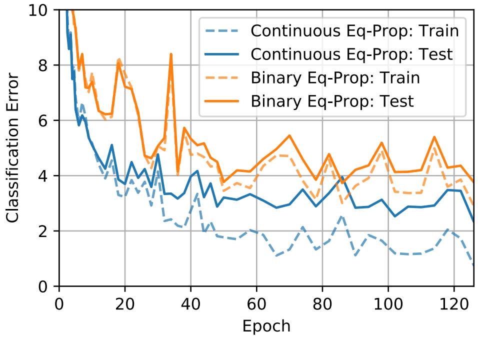


Figure 5: Parallel Coordinates plot for the hyperparameter search for the 1-layer MNIST network. See Figure 4 caption for explanation of the plot. The hyperparameter search aimed to minimize the validation-set error after 1 epoch. **Left**: The search using scheduled step-size / predictive coding. **Right**: The search using OSA and fixed predictive-coding. Because these behaved similarly, we used the OSA + predictive coding (right) which had fewer parameters, with the optimal parameters found here model for Figure 3.

We also experimented with a deeper network,with 3 hidden layers of 500 units. The following plot shows the learning curve of Equilibrium Prop. The "binary" network is run with OSA, with parameters $\lambda = 0.771, \bar{\nu} = 0.686$ found in a hyperparameter search.



A final note is that it is possible to get our spiking network to perform on par with Equilibrium Prop *without* increasing the number of convergence steps (T^+ and T^-). The trick is to save a "checkpoint" of the state of the neurons (including the states of all encoders and decoders), at the end of the negative phase (see the "splitstream" parameter in the code). We then allow both the negative phase and the positive phase to proceed independently from this checkpoint for T^+ steps, to achieve the s^- and s^+ used in the update rule in Equation 4. This results in a much lower variance gradient estimate because the noise due to the internal states of the encoders/decoders cancels out in the constrastive update. We do not use this in any of our experiments because the notion of saving a state "checkpoint" which you can return to is biologically unrealistic.