The Eighty Five Percent Rule for Optimal Learning

Robert C. Wilson[a,1], Amitai Shenhav[b,c], Mark Straccia[d], and Jonathan D. Cohen[e]

[a]Department of Psychology and Cognitive Science Program, University of Arizona
[b]Cognitive, Linguistic, & Psychological Sciences, Brown University
[c]Brown Institute for Brain Science, Brown University
[d]Department of Psychology, UCLA
[e]Princeton Neuroscience Institute, Princeton University
[1]To whom correspondence should be addressed. E-mail: bob@arizona.edu

Abstract

Researchers and educators have long wrestled with the question of how best to teach their clients be they human, animal or machine. Here we focus on the role of a single variable, the difficulty of training, and examine its effect on the rate of learning. In many situations we find that there is a sweet spot in which training is neither too easy nor too hard, and where learning progresses most quickly. We derive conditions for this sweet spot for a broad class of learning algorithms in the context of binary classification tasks, in which ambiguous stimuli must be sorted into one of two classes. For all of these gradient-descent based learning algorithms we find that the optimal error rate for training is around 15.87% or, conversely, that the optimal training accuracy is about 85%. We demonstrate the efficacy of this 'Eighty Five Percent Rule' for artificial neural networks used in AI and biologically plausible neural networks thought to describe human and animal learning.

Introduction

When we learn something new, like a language or musical instrument, we often seek challenges at the edge of our competence – not so hard that we are discouraged, but not so easy that we get bored. This simple intuition, that there is a sweet spot of difficulty, a 'Goldilocks zone' [1], for motivation and learning is at the heart of modern teaching methods [2] and is thought to account for differences in infant attention between more and less learnable stimuli [1]. In the animal learning literature it is the intuition behind shaping [3] and fading [4], whereby complex tasks are taught by steadily increasing the difficulty of a training task. It is also observable in the nearly universal 'levels' feature in video games, in which the player is encouraged, or even forced, to a higher level of difficulty once a performance criterion has been achieved. Similarly in machine learning, steadily increasing the difficulty of training has proven useful for teaching large scale neural networks in a variety of tasks [5,6], where it is known as 'Curriculum Learning' [7] and 'Self-Paced Learning' [8].

Despite this long history of empirical results, it is unclear *why* a particular difficulty level may be beneficial for learning nor what that optimal level might be. In this paper we address this issue of optimal training difficulty for a broad class of learning algorithms in the context of binary classification tasks, where ambiguous stimuli must be classified into one of two classes (e.g. cat or dog).

In particular, we focus on the class of gradient-descent based learning algorithms. In these algorithms, parameters of the model (e.g. the weights in a neural network) are adjusted based on feedback in such a way as to reduce the average error rate over time [9]. That is, these algorithms descend the gradient of error rate as a function of model parameters. Such gradient-descent learning forms the basis of many algorithms in AI, from single-layer perceptrons to deep neural networks [10], and provides a quantitative description of human and animal learning in a variety of situations, from perception [11], to motor control [12] to reinforcement learning [13]. For these algorithms, we provide a general result for the optimal difficulty in terms of a target error rate for training. Under fairly mild assumptions this optimal error rate is around 15.87%, a number that varies slightly depending on the noise in the learning process. We show theoretically that training at this optimal difficulty can lead to exponential improvements in the rate of learning. Finally, we demonstrate the applicability of the Eighty Five Percent Rule in two cases: a simple artificial neural network, the single-layer perceptron [14], and a more complex biologically plausible network thought to describe human and animal perceptual learning [11].

Results

Optimal training difficulty for binary classification tasks

In a standard binary classification task, a human, animal or machine 'agent' make binary decisions about simple stimuli. For example, in the classic Random Dot Motion paradigm from Psychology and Neuroscience [15,16], stimuli consist of a patch of moving dots – most moving randomly but a small fraction moving coherently either to the left or the right – and participants must decide in which direction the coherent dots are moving. A major factor in determining the difficulty of this perceptual decision is the fraction of coherently moving dots, which can be manipulated by the experimenter to achieve a fixed error rate during training using a procedure known as 'staircasing' [17].

We assume that agents make their decision on the basis of a scalar, subjective decision variable, h, which is computed from a stimulus that can be represented as a vector $\mathbf{x}$ (e.g. the direction of motion of all dots)

$$h = \Phi(\mathbf{x}, \phi) \tag{1}$$

where $\Phi(\cdot)$ is a function of the stimulus and (tunable) parameters ϕ. We assume that this transformation of stimulus, $\mathbf{x}$ into the subjective decision variable h yields a noisy representation of the true decision variable, Δ (e.g. the fraction of dots moving left). That is, we write

$$h = \Delta + n \tag{2}$$

where the noise, n, arises due to the imperfect representation of the decision variable. We further assume that this noise, n, is random and sampled from a zero-mean Gaussian distribution with standard deviation σ (Figure 1A).

If the decision boundary is set to 0, such that the model chooses option A when $h > 0$, option B when $h < 0$ and randomly when $h = 0$, then the noise in the representation of the decision variable leads to errors with probability

$$ER = \int_{-\infty}^{0} p(h|\Delta, \sigma) dh = F(-\Delta/\sigma) = F(-\beta\Delta) \tag{3}$$

where $F(x)$ is the cumulative density function of the standardized noise distribution, $p(x) = p(x|0, 1)$, and $\beta = 1/\sigma$ quantifies the precision of the representation of Δ and the agent's skill at the task. As shown in Figure 1B, this error rate decreases as the decision gets easier (Δ increases) and as the agent becomes more accomplished at the task (β increases).

The goal of learning is to tune the parameters ϕ such that the subjective decision variable, h, is a better reflection of the true decision variable, Δ. That is, the model should aim to adjust the parameters ϕ so as to decrease the magnitude of the noise σ or, equivalently, increase the precision β. One way to achieve this tuning is to adjust the parameters using gradient descent on the error rate, i.e. changing the parameters over time t according to

$$\frac{d\phi}{dt} = -\eta \nabla_{\phi} ER \tag{4}$$

where η is the learning rate and $\nabla_{\phi} ER$ is the derivative of the error rate with respect to parameters ϕ. This gradient can be written in terms of the precision, β, as

$$\nabla_{\phi} ER = \frac{\partial ER}{\partial \beta} \nabla_{\phi} \beta \tag{5}$$

Note here that only the first term on the right hand side of equation 5 depends on the difficulty Δ, while the second describes how the precision changes with ϕ. This means that the optimal difficulty for training is the value of the decision variable Δ^* that maximizes $\partial ER/\partial \beta$.

In terms of the decision variable, the optimal difficulty changes as a function of precision (Figure 1C) meaning that the difficulty of training must be adjusted online according to

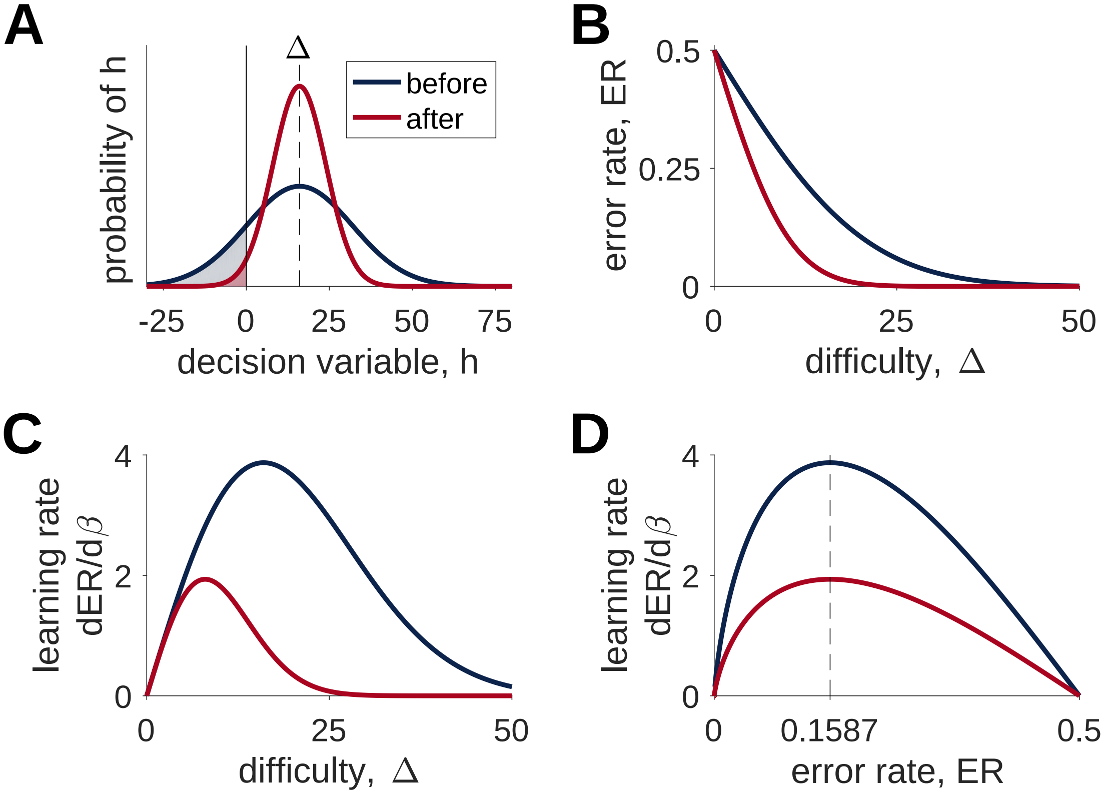

Figure 1: Illustration of the model. (A) Distributions over decision variable h given a particular difficulty, $\Delta = 16$, with lower precision before learning and higher precision after learning. The shaded regions corresponds to the error rate – the probability of making an incorrect response at each difficulty. (B) The error rate as a function of difficulty before and after learning. (C) The derivative that determines the rate of learning as a function of difficulty before and after learning showing that the optimal difficulty for learning is lower after learning than before. (D) The same derivative as in (C) re-plotted as a function of error rate showing that the optimal error rate (at 15.87%) is the same both before and after learning.

the skill of the agent. However, by using the monotonic relationship between Δ and ER (Figure 1B) it is possible to express the optimal difficulty in terms of the error rate, ER^* (Figure 1D). Expressed this way, the optimal difficulty is constant as a function of precision, meaning that optimal learning can be achieved by clamping the error rate during training at a fixed value, which, for Gaussian noise is

$$ER^* = \frac{1}{2}\left(1 - \mathrm{erf}\left(\frac{1}{\sqrt{2}}\right)\right) \approx 0.1587 \tag{6}$$

Dynamics of learning

While the previous analysis allows us to calculate the error rate that maximizes the rate of learning, it does not tell us how much faster learning occurs at this optimal error rate. In this section we address this question by comparing learning at the optimal error rate with learning at a fixed, but potentially suboptimal error rate, ER_f, and a fixed difficulty, Δ_f. In both cases, gradient-descent based updating of the parameters, ϕ, (Equation 4) implies that the precision β evolves in a similar manner, i.e.

$$\frac{d\beta}{dt} = -\eta \frac{\partial ER}{\partial \beta} \tag{7}$$

Fixed error rate

As shown in the Methods, integrating Equation 7 for fixed error rate gives

$$\beta(t) = \sqrt{\beta_0^2 + 2\eta K_f (t - t_0)} \tag{8}$$

where t_0 is the initial time point, β_0 is the initial value of β and K_f is the following function of the training error rate

$$K_f = -F^{-1}(ER_f)p(F^{-1}(ER_f)) \tag{9}$$

Thus, for fixed training error rate the precision grows as the square root of time with the exact rate determined by K_f which depends on both the training error rate and the noise distribution.

Fixed decision variable

When the decision variable is fixed, Δ_f, integrating equation 7 is more difficult and the solution depends more strongly on the distribution of the noise. In the case of Gaussian noise, there is no closed form solution for β. However, as shown in the Methods, an approximate form can be derived at long times where we find that β grows as

$$\beta(t) \propto \sqrt{\log t} \tag{10}$$

i.e. exponentially slower than equation 34.

Simulations

To demonstrate the applicability of the Eighty Five Percent Rule we simulated the effect of training accuracy on learning in two cases. From AI we consider the classic Perceptron [14], a simple artificial neural network that has been used in a variety of applications from handwriting recognition [18] to natural language processing [19]. From computational neuroscience we consider the model of Law and Gold [11], that accounts for both the behavior and neural firing properties of monkeys learning the Random Dot Motion task. In both cases we see that learning is maximized when training occurs at 85% accuracy.

Perceptron

The Perceptron is a classic one-layer neural network model that learns to map multidimensional stimuli $\mathbf{x}$ onto binary labels, y via a linear threshold process [14]. To implement this mapping, the Perceptron first computes the decision variable h as

$$h = \mathbf{w} \cdot \mathbf{x} \tag{11}$$

where $\mathbf{w}$ are the weights of the network, and then assigns the label according to

$$y = \begin{cases} 1 & h > 0 \\ 0 & h \leq 0 \end{cases} \tag{12}$$

The weights, $\mathbf{w}$, which constitute the parameters of the model, are updated based on feedback about the true label t by a the learning rule,

$$\mathbf{w} \leftarrow \mathbf{w} + (t - y)\mathbf{x} \tag{13}$$

This learning rule implies that the Perceptron only updates its weights when the predicted label y does not match the actual label t – that is, the Perceptron only learns when it makes mistakes. Naïvely then, one might expect that optimal learning would involve maximizing the error rate. However, because Equation 13 is actually a gradient descent based rule (e.g. Chapter 39 in [20]), the analysis of the previous sections applies and the optimal error rate for training is 15.87%.

To test this prediction we simulated the Perceptron learning rule for a range of training error rates between 0.01 and 0.5 in steps of 0.01 (1000 simulations per error rate). The degree of learning was captured by the precision β (see Methods). As predicted by the theory, the network learns most effectively when trained at the optimal error rate (Figure 2A) and the dynamics of learning are well described, up to a scale factor, by Equation 34 (Figure 2B).

Biologically plausible model of perceptual learning

To demonstrate how the Eighty Five Percent Rule might apply to learning in biological systems, we simulated the Law and Gold model of perceptual learning [11]. This model has been shown to capture the long term changes in behavior, neural firing and synaptic weights as monkeys learn to perform the Random Dot Motion task.

Specifically, the model assumes that monkeys make the perceptual decision between left and right on the basis of neural activity in area MT – an area in the dorsal visual stream that

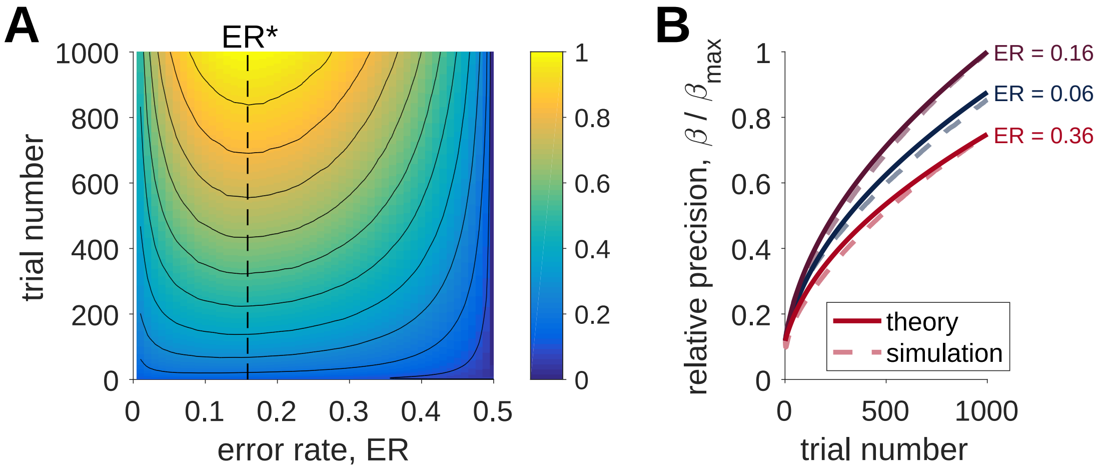

Figure 2: The Eighty Five Percent Rule applied to the Perceptron. (A) The relative precision, β/β_{max}, as a function of training error rate and training duration. Training at the optimal error rate leads to the fastest learning throughout. (B) The dynamics of learning agree well with the theory.

is known to represent motion information [15]. In the Random Dot Motion task, neurons in MT have been found to respond to both the direction θ and coherence COH of the dot motion stimulus such that each neuron responds most strongly to a particular 'preferred' direction and that the magnitude of this response increases with coherence. This pattern of firing is well described by a simple set of equations (see Methods) and thus the noisy population response, $\mathbf{x}$, to a stimulus of arbitrary direction and coherence is easily simulated.

From this MT population response, Law and Gold proposed that animals construct a decision variable in a separate area of the brain (lateral interparietal area, LIP) as the weighted sum of activity in MT; i.e.

$$h = \mathbf{w} \cdot \mathbf{x} + \epsilon \tag{14}$$

where $\mathbf{w}$ are the weights between MT and LIP neurons and ϵ is random neuronal noise that cannot be reduced by learning. The presence of this irreducible neural noise is a key difference between the Law and Gold model (Equation 14) and the Perceptron (Equation 11) as it means that no amount of learning can lead to perfect performance. However, as shown in the Methods section, the presence of irreducible noise does not change the optimal accuracy for learning which is still 85%.

Another difference between the Perceptron and the Law and Gold model is the form of the learning rule. In particular, weights are updated according to a reinforcement learning rule based on a reward prediction error

$$\delta = r - E[r] \tag{15}$$

where r is the reward presented on the current trial (1 for a correct answer, 0 for an incorrect answer) and $E[r]$ is the predicted reward

$$E[r] = \frac{1}{1 + \exp(-B|h|)} \tag{16}$$

where B is a proportionality constant that is estimated online by the model (see Methods). Given the prediction error, the model updates its weights according to

$$\mathbf{w} \leftarrow \mathbf{w} + \eta C \delta \mathbf{x} \tag{17}$$

where C is the choice (-1 for left, +1 for right) and η is the learning rate. Despite the superficial differences with the Perceptron learning rule (Equation 13) the Law and Gold model still implements gradient descent on the error rate [13] and learning should be optimized at 85%.

To test this prediction we simulated the model at a variety of different target training error rates. Each target training rate was simulated 100 times with different parameters for the MT neurons (see Methods). The precision, β, of the trained network was estimated by fitting simulated behavior of the network on a set of test coherences that varied logarithmically between 1 and 100%. As shown in Figure 3A the precision after training is well described (up to a scale factor) by the theory. In addition, in Figure 3B, we show the expected difference in behavior - in terms of psychometric choice curves - for three different training error rates. While these differences are small, they are large enough that they could be distinguished experimentally.

Discussion

In this paper we considered the effect of training accuracy on learning in the case of binary classification tasks and gradient-descent-based learning rules. We found that the rate of learning is maximized when the difficulty of training is adjusted to keep the training accuracy at around 85%. We showed that training at the optimal accuracy proceeds exponentially faster than training at a fixed difficulty. Finally we demonstrated the efficacy of the Eighty Five Percent Rule in the case of artificial and biologically plausible neural networks.

Our results have implications for a number of fields. Perhaps most directly, our findings move towards a theory for identifying the optimal environmental settings in order to maximize the rate of gradient-based learning. Thus the Eighty Five Percent Rule should apply to a wide range of machine learning algorithms including multilayered feedforward and recurrent neural networks (e.g. including 'deep learning' networks using backpropagation [9], Boltzmann machines [21], reservoir computing networks [22, 23]), as well as Perceptrons (as we showed here). In addition the Eighty Five Percent Rule accords with the informal intuition of many experimentalists that participant engagement is often maximized when performance is maintained around 85% [24]. Indeed it is notable that staircasing procedures (that aim to titrate difficulty such that error rate is fixed during learning) are commonly designed to produce about 85% accuracy [17]. Despite the prevalence of this intuition, to the best of our knowledge no formal theoretical work has addressed the effect of training accuracy on learning, a test of which is an important direction for future work.

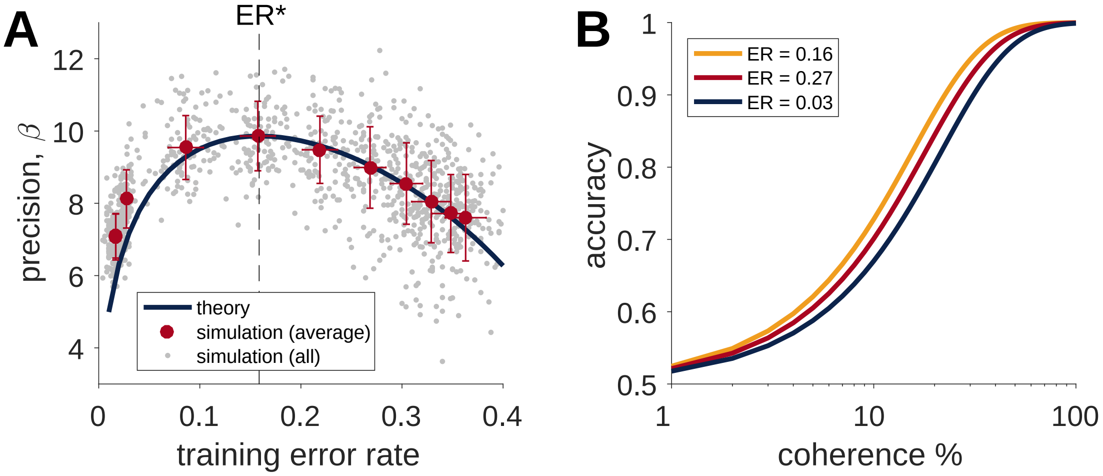

Figure 3: The Eighty Five Percent Rule optimizes learning in the Law and Gold model of perceptual learning. (A) Precision of the trained network as function of training error rate. Grey dots represent the results of individual simulations – note that the empirical error rate on each run often differs slightly from the target error rate due to noise. Red dots correspond to the average precision and empirical error rate for each target error rate (error bars ± standard deviation in both measures). (B) Accuracy as a function of coherence for the network trained at three different error rates corresponding to near optimal ($ER = 0.16$), too high ($ER = 0.27$) and too low ($ER = 0.03$).

More generally, our work closely to the Region of Proximal Learning and Desirable Difficulty frameworks in education [25–27] and Curriculum Learning and Self-Paced Learning [7, 8] in computer science. These related, but distinct, frameworks propose that people and machines should learn best when training tasks involve just the right amount of difficulty. In the Desirable Difficulties framework, the difficulty in the task must be of a 'desirable' kind, such as spacing practice over time, that promotes learning as opposed to an undesirable kind that does not. In the Region of Proximal Learning framework, which builds on early work by Piaget [28] and Vygotsky [29], this optimal difficulty is in a region of difficulty just beyond the person's current ability. Curriculum and Self-Paced Learning in computer science build on similar intuitions, that machines should learn best when training examples are presented in order from easy to hard. In practice, the optimal difficulty in all of these domains is determined empirically and is often dependent on many factors [30]. In this context, our work offers a way of deriving the desired difficulty and the region of proximal learning in the special case of binary classification tasks and gradient-descent learning rules. As such our work represents the first step towards a more mathematical instantiation of these theories, although it remains to be generalized to a broader class of circumstances, such as multi-choice tasks and different learning algorithms.

With regard to different learning algorithms, it is important to note that not all models will exhibit a sweet spot of difficulty for learning. As an example, consider how a Bayesian learner would infer parameters ϕ by computing the posterior distribution given past stimuli,

$\mathbf{x}_{1:t}$, and labels, $y_{1:t}$,

$$\begin{aligned} p(\phi|\mathbf{x}_{1:t}, y_{1:t}) &\propto p(y_{1:t}|\phi, \mathbf{x}_{1:t})p(\phi) \\ &= \prod_{i=1}^{t} p(y_i|\phi, \mathbf{x}_i)p(\phi) \end{aligned} \tag{18}$$

where the last line holds when the label depends only on the current stimulus. Clearly this posterior distribution over parameters is independent of the ordering of the trials meaning that a Bayesian learner would learn equally well if hard or easy examples are presented first. This is not to say that Bayesian learners cannot benefit from carefully constructed training sets, but that for a given set of training items the order of presentation has no bearing on what is ultimately learned. This contrasts markedly with gradient-based algorithms, many of which try to approximate the maximum *a posteriori* solution of a Bayesian model, whose training is order dependent and whose learning is optimized with $\partial ER/\partial\beta$.

Finally, we note that our analysis for maximizing the gradient, $\partial ER/\partial\beta$, not only applies to learning but to any process that affects the precision of neural representations, such as attention, engagement, or more generally cognitive control [31, 32]. For example, attention is known to improve the precision with which sensory stimuli are represented in the brain, e.g. [33]. If exerting control leads to a change in precision of $\delta\beta$, then the change in error rate associated with exerting this control is

$$\delta ER = \frac{\partial ER}{\partial \beta}\delta\beta \tag{19}$$

This predicts that the benefits of engaging cognitive control should be maximized when $\partial ER/\partial\beta$ is maximized, that is at ER^*. More generally this relates to the Expected Value of Control theory [31, 32, 34] which suggests that the learning gradient, $\partial ER/\partial\beta$, is monitored by control-related areas of the brain such as anterior cingulate cortex.

Along similar lines, our work points to a mathematical theory of the state of 'Flow' [35]. This state, 'in which an individual is completely immersed in an activity without reflective self-consciousness but with a deep sense of control', is thought to occur most often when the demands of the task are well matched to the skills of the participant. This idea of balance between skill and challenge was captured originally with a simple conceptual diagram (Figure 4) with two other states: 'anxiety' when challenge exceeds skill and 'boredom' when skill exceeds challenge. These three qualitatively different regions (flow, anxiety and boredom) arise naturally in our model. Identifying the precision, β, with the level of skill and the level challenge with the inverse of true decision variable, $1/\Delta$, we see that when challenge equals skill, flow is associated with a high learning rate and accuracy, anxiety with low learning rate and accuracy and boredom with high accuracy but low learning rate (Figure 4B and C). Intriguingly, recent work by Vuorre and Metcalfe, has found that subjective feelings of Flow peaks on tasks that are subjectively rated as being of intermediate difficulty [36]. In addition work on learning to control brain computer interfaces finds that subjective, self-reported measures of 'optimal difficulty', peak at a difficulty associated with maximal learning not at a difficulty associated with optimal decoding of neural activity [37]. Going forward, it will be interesting to test whether these subjective measures of engagement peak at the point of maximal learning gradient, which for binary classification tasks is 85%.

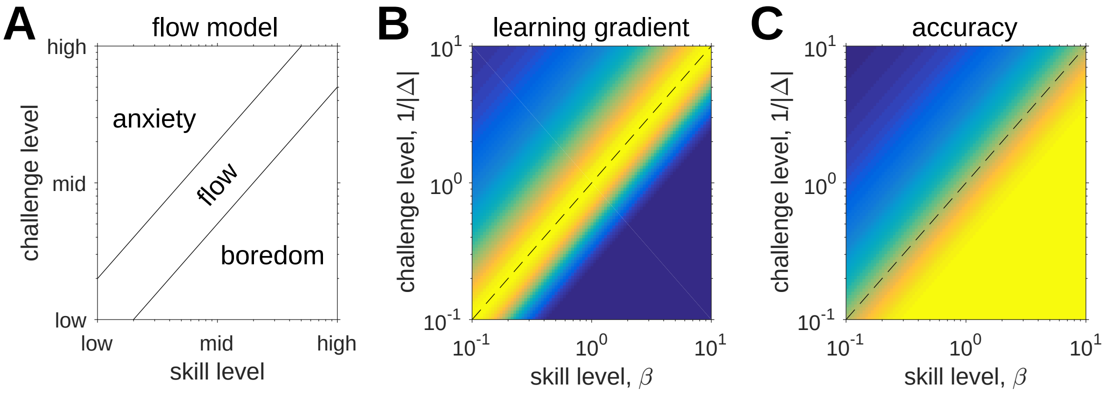

Figure 4: Proposed relationship between the Eighty Five Percent Rule and Flow. (A) Original model of flow as a state that is achieved when skill and challenge are well balanced. Normalized learning rate, $\partial ER/\partial\beta$, (B) and accuracy (C) as a function of skill and challenge suggests that flow corresponds to high learning and accuracy, boredom corresponds to low learning and high accuracy, while anxiety is associated with low learning and low accuracy.

Methods

Optimal error rate for learning

In order to compute the optimal difficulty for training, we need to find the value of Δ that maximizes the learning gradient, $\partial ER/\partial\beta$. From Equation 3 we have

$$\frac{\partial ER}{\partial \beta} = \Delta p(-\beta\Delta) \tag{20}$$

From here the optimal difficulty, Δ^*, can be found by computing the derivative of the gradient with respect to Δ, i.e.

$$\begin{aligned}\frac{\partial}{\partial \Delta}\frac{\partial ER}{\partial \beta} &= -\frac{\partial}{\partial \Delta}\left(\Delta p(-\beta\Delta)\right)\\ &= -p(-\beta\Delta) + \beta\Delta\left.\frac{\partial p(x)}{\partial x}\right|_{x=-\beta\Delta}\end{aligned} \tag{21}$$

Setting this derivative equal to zero gives us the following expression for the optimal difficulty, Δ^*, and error rate, ER^*

$$\beta\Delta^* = \frac{p(-\beta\Delta^*)}{p'(-\beta\Delta^*)} \quad \text{and} \quad ER^* = F(-\beta\Delta^*) \tag{22}$$

where $p'(x)$ denotes the derivative of $p(x)$ with respect to x. Because β and Δ^* only ever appear together in these expressions, equation 22 implies that $\beta\Delta^*$ is a constant. Thus, while the optimal difficulty, Δ^*, changes as a function of precision (Figure 1C), the optimal training error rate, ER^* does not (Figure 1D). That is, training with the error rate clamped at ER^* is guaranteed to maximize the rate of learning.

The exact value of ER^* depends on the distribution of noise, n, in Equation 2. In the case of Gaussian noise, we have

$$p(x) = \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{x^2}{2}\right) \tag{23}$$

which implies that

$$\frac{p(x)}{p'(x)} = -\frac{1}{x} \tag{24}$$

and that the optimal difficulty is

$$\Delta^* = \beta^{-1} \tag{25}$$

Consequently the optimal error rate for Gaussian noise is

$$ER^* = \frac{1}{2}\left(1 - \text{erf}\left(\frac{1}{\sqrt{2}}\right)\right) \approx 0.1587 \tag{26}$$

Similarly for Laplacian noise ($p(x) = \frac{1}{2}\exp(-|x|)$) and Cauchy noise ($p(x) = (\pi(1+x^2))^{-1}$) we have optimal error rates of

$$\begin{aligned} ER^*_{Laplace} &= \frac{1}{2}\exp(-1) \approx 0.1839 \\ ER^*_{Cauchy} &= \frac{1}{\pi}\arctan(-1) + \frac{1}{2} = 0.25 \end{aligned} \tag{27}$$

Optimal learning with endogenous noise

The above analyses for optimal training accuracy also applies in the case where the decision variable, h, is corrupted by endogenous, irreducible noise, ϵ, in addition to representation noise, n, that can be reduced by learning; i.e.

$$h = \Delta + n + \epsilon \tag{28}$$

In this case we can split the overall precision, β, into two components, one based on representational uncertainty that can be reduced, β_n, and another based on endogenous uncertainty that cannot, β_ϵ. For Gaussian noise, these precisions are related to each other by

$$\frac{1}{\beta^2} = \frac{1}{\beta_n^2} + \frac{1}{\beta_\epsilon^2} \tag{29}$$

More generally, the precisions are related by some function, G, such that $\beta = G(\beta_n, \beta_\epsilon)$. Since only n can be reduced by learning, it makes sense to perform gradient descent on β_n such that the learning rule should be

$$\begin{aligned} \frac{d\beta_n}{dt} &= -\eta \frac{\partial ER}{\partial \beta_n} \\ &= -\eta \frac{\partial ER}{\partial \beta} \frac{\partial \beta}{\partial \beta_n} \end{aligned} \tag{30}$$

Note that $\partial\beta/\partial\beta_n$ is independent of Δ so maximizing learning rate w.r.t. Δ means maximizing $\partial ER/\partial\beta$ as before. This implies that the optimal training difficulty will be the same, e.g. 85% for Gaussian noise, regardless whether endogenous noise is present or not.

Dynamics of learning

To calculate the dynamics of learning we need to integrate equation 7 over time. This, of course depends on the learning gradient, $\partial ER/\partial\beta$, which varies depending on the noise and whether the error rate or the true decision variable is fixed during training.

Fixed error rate

In this case we fix the error rate during training to ER_f. This implies that the difficulty should change over time according to

$$\Delta(t) = -\frac{1}{\beta(t)} F^{-1}(ER_f) \tag{31}$$

where $F^{-1}(\cdot)$ is the inverse cdf. This implies that β evolves over time according to

$$\begin{aligned} \frac{d\beta}{dt} &= -\eta \frac{\partial ER}{\partial \beta} \\ &= \eta \Delta(t) p(-\beta\Delta(t)) \\ &= -\frac{\eta}{\beta(t)} F^{-1}(ER_f) p(F^{-1}(ER_f)) \\ &= \frac{\eta K_f}{\beta(t)} \end{aligned} \tag{32}$$

where we have introduced K_f as

$$K_f = -F^{-1}(ER_f) p(F^{-1}(ER_f)) \tag{33}$$

Integrating equation 32 and solving for $\beta(t)$ we get

$$\beta(t) = \sqrt{\beta_0^2 + 2\eta K_f (t - t_0)} \tag{34}$$

where t_0 is the initial time point, and β_0 is the initial value of β. Thus, for fixed error rate the precision grows as the square root of time with the rate determined by K_f which depends on both the training error rate and the noise distribution. For the optimal error rate we have, $K_f = p(-1)$.

Fixed decision variable

In this case the true decision variable is fixed at Δ_f and the error rate varies as a function of time. In this case we have

$$\frac{d\beta}{dt} = -\eta \frac{\partial ER}{\partial \beta} = \Delta_f p(-\beta\Delta_f) \tag{35}$$

Formally, this can be solved as

$$\int_{\beta_0}^{\beta} \frac{1}{p(-\beta\Delta_f)} d\beta = \Delta_f (t - t_0) \tag{36}$$

However, the exact form for $\beta(t)$ will depend on $p(x)$.

In the Gaussian case we cannot derive a closed form expression for $\beta(t)$. The closest we can get is to write

$$\int_0^{\frac{\beta\Delta_f}{\sqrt{2}}} \exp(x^2)dx = \int_0^{\frac{\beta_0\Delta_f}{\sqrt{2}}} \exp(x^2)dx + \frac{\Delta^2}{2\sqrt{\pi}}(t-t_0) \tag{37}$$

For long times, and large β, we can write

$$\int_0^{\frac{\beta\Delta_f}{\sqrt{2}}} \exp(x^2)dx < \exp\left(\frac{\beta^2\Delta_f^2}{2}\right) \tag{38}$$

which implies that for long times β grows slower than $\sqrt{\log t}$, which is exponentially slower than the fixed error rate case.

In contrast to the Gaussian case, the Laplacian case lends itself to closed form analysis and we can derive the following expression for β

$$\beta = \frac{1}{\Delta_f}\log\left(\exp(\beta_0\Delta_f) + \frac{1}{2}\eta\Delta_f^2(t-t_0)\right) \tag{39}$$

Again this shows logarithmic dependence on t indicating that learning is much slower with a fixed difficulty.

In the case of Cauchy noise we can compute the integral in equation and find that β is the root of the following equation

$$\frac{\Delta_f}{3}\beta^3 + \beta = \frac{\Delta_f}{3}\beta_0^3 + \beta_0 + \frac{\Delta_f}{\pi}(t-t_0) \tag{40}$$

For long training times this implies that β grows as the cube root of t. Thus in the Cauchy case, while the rate of learning is still greatest at the optimal difficulty, the improvement is not as dramatic as in the other cases.

Application to the Perceptron

To implement the Perceptron example, we assumed that true labels t were generated by a 'Teacher Perceptron' [38] with normalized weight vector, $\mathbf{e}$. Learning was quantified by decomposing the learned weights $\mathbf{w}$ into two components: one proportional to $\mathbf{e}$ and a second orthogonal to $\mathbf{e}$, i.e.

$$\mathbf{w} = |\mathbf{w}|\left(\mathbf{e}\cos\theta + \mathbf{e}_\perp \sin\theta\right) \tag{41}$$

where θ is the angle between $\mathbf{w}$ and $\mathbf{e}$, and $\mathbf{e}_\perp$ is the unit vector perpendicular to $\mathbf{e}$ in the plane defined by $\mathbf{e}$ and $\mathbf{w}$. This allows us to write the decision variable h in terms of signal and noise components as

$$\begin{aligned} h &= |\mathbf{w}|\left((\mathbf{e}\cdot\mathbf{x})\cos\theta + (\mathbf{e}_\perp\cdot\mathbf{x})\sin\theta\right) \\ &= \underbrace{|\mathbf{w}|(2t-1)\Delta\cos\theta}_{\text{signal}} + \underbrace{|\mathbf{w}|(\mathbf{e}_\perp\cdot\mathbf{x})\sin\theta}_{\text{noise}} \end{aligned} \tag{42}$$

where the difficulty $\Delta = |\mathbf{e} \cdot \mathbf{x}|$ is the distance between $\mathbf{x}$ and the decision boundary, and the $(2t-1)$ term simply controls which side of the boundary $\mathbf{x}$ is on. This implies that the precision β is proportional to $\cot\theta$, with a constant of proportionality determined by the dimensionality of $\mathbf{x}$.

In the case where the observations $\mathbf{x}$ are sampled from distributions that obey the central limit theorem, then the noise term is approximately Gaussian implying that the optimal error rate for training the Perceptron, $ER^* = 15.87\%$.

To test this prediction we simulated the Perceptron learning rule for a range of training error rates between 0.01 and 0.5 in steps of 0.01 (1000 simulations per error rate). Stimuli, $\mathbf{x}$, were 100 dimensional and independently sampled from a Gaussian distribution with mean 0 and variance 1. Similarly, the true weights $\mathbf{e}$ were sampled from a mean 0, variance 1 Gaussian. To mimic the effect of a modest degree of initial training, we initialized the weight vector $\mathbf{w}$ randomly with the constraint that $|\theta| < 1.6\pi$. The difficulty Δ was adjusted on a trial-by-trial basis according to

$$\Delta = F^{-1}(ER)\lambda \tan\theta \tag{43}$$

which ensures that the training error rate is clamped at ER. The degree of learning was captured by the precision β.

Application to Law and Gold model

The model of perceptual learning follows the exposition in Law and Gold [11]. To aid comparison with that paper we retain almost all of their notation, with the two exceptions being their β parameter which we rename as B to avoid confusion with the precision and their learning rate parameter α which we write as η.

MT neuron activity

Following Law and Gold [11], the average firing rate of an MT neuron, i, in response to a moving dot stimulus with direction θ and coherence COH is

$$m_i = T(k_i^0 + COH(k_i^n + (k_i^p - k_i^n)f(\theta|\Theta_i))) \tag{44}$$

where T is the duration of the stimulus, k_i^0 is the response of neuron i to a zero-motion coherence stimulus, k_i^p is the response to a stimulus moving in the preferred direction and k_i^n is the response to a stimulus in the null direction. $f(\theta|\Theta_i)$ is the tuning curve of the neuron around its preferred direction Θ_i

$$f(\theta|\Theta_i) = \exp\left(-\frac{(\theta - \Theta_i)^2}{2\sigma_\theta^2}\right) \tag{45}$$

where σ_θ (= 30 degrees) is the width of the tuning curve which is assumed to be identical for all neurons.

Neural activity on each trial was assumed to be noisily distributed around this mean firing rate. Specifically the activity, x_i, of each neuron is given by a rectified (to ensure $x_i > 0$) sample from a Gaussian with mean m_i and variance v_i

$$v_i = \phi_i m_i \tag{46}$$

where ϕ_i is the Fano factor of the neuron.

Thus each MT neuron was characterized by five free parameters. These free parameters were sampled randomly for each neuron such that $\theta_i \sim U(-180, 180)$, $k_i^0 \sim U(0, 20)$, $k_i^p \sim U(0, 50)$, $k_i^n \sim U(-k_i^0, 0)$ and $\phi_i \sim U(1, 5)$. Note that k_i^n is set between $-k_i^0$ and 0 to ensure that the minimum average firing rate never dips below zero. Each trial was defined by three task parameters: $T = 1$ second, $\Theta = \pm 90$ degrees and COH which was adjusted based on performance to achieve a fixed error rate during training (see below). As in the original paper, the number of neurons was set to 7200 and the learning rate, η was 10^{-7}.

Computing predicted reward

The predicted reward $E[r]$ was computed according to equation 16. In line with Law and Gold (Supplemental Figure 2 in [11]), the proportionality constant B was computed using logistic regression on the accuracy and absolute value of the decision variable, $|h|$, from last L trials, where $L = \min(300, t)$.

Weight normalization

In addition to the weight update rule (Equation 17), weights were normalized after each update to keep the sum of the squared weights, $\sum_i w_i^2 = w_{amp}$ a constant (=0.02). While this normalization has only a small overall effect (see Supplementary Material in [11]), we replicate this weight normalization here for consistency with the original model.

Adjusting coherence to achieve fixed training difficulty

To initialize the network, the first 50 trials of the simulation had a fixed coherence $COH = 0.9$. After this initialization period, the coherence was adjusted according to the difference between the target accuracy, A_{target}, and actual accuracy in the last L trials, A_L, where $L = \min(300, t)$. Specifically, the coherence on trial t was set as

$$COH_t = \frac{1}{1 + \exp(-\Gamma_t)} \tag{47}$$

where Γ_t was adjusted according to

$$\Gamma_{t+1} = \Gamma_t + d\Gamma(A_{target} - A_L) \tag{48}$$

and $d\Gamma$ was 0.1.

Estimating precision parameter post training

To estimate the post-training precision parameter, β, we simulated behavior of the trained network on a set of 20 logarithmically spaced coherences between 10^{-3} to 1. Behavior at each coherence was simulated 100 times and learning was disabled during this testing phase. The precision parameter, β, was estimated using logistic regression between accuracy on each trial (0 or 1) and coherence; i.e.,

$$ACC \sim \frac{1}{1 + \exp(-\beta \times COH)} \tag{49}$$

References

1. Celeste Kidd, Steven T Piantadosi, and Richard N Aslin. The goldilocks effect: Human infants allocate attention to visual sequences that are neither too simple nor too complex. *PloS one*, 7(5):e36399, 2012.

2. Janet Metcalfe. Metacognitive judgments and control of study. *Current Directions in Psychological Science*, 18(3):159–163, 2009.

3. BF Skinner. The behavior of organisms: An experimental analysis. new york: D. appleton-century company, 1938.

4. Douglas H Lawrence. The transfer of a discrimination along a continuum. *Journal of Comparative and Physiological Psychology*, 45(6):511, 1952.

5. J L Elman. Learning and development in neural networks: the importance of starting small. *Cognition*, 48(1):71–99, Jul 1993.

6. Kai A Krueger and Peter Dayan. Flexible shaping: How learning in small steps helps. *Cognition*, 110(3):380–394, 2009.

7. Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. Curriculum learning. In *Proceedings of the 26th annual international conference on machine learning*, pages 41–48. ACM, 2009.

8. M Pawan Kumar, Benjamin Packer, and Daphne Koller. Self-paced learning for latent variable models. In *Advances in Neural Information Processing Systems*, pages 1189–1197, 2010.

9. David E Rumelhart, Geoffrey E Hinton, Ronald J Williams, et al. Learning representations by back-propagating errors. *Cognitive modeling*, 5(3):1, 1988.

10. Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *Nature*, 521(7553):436–444, 2015.

11. Chi-Tat Law and Joshua I Gold. Reinforcement learning can account for associative and perceptual learning on a visual-decision task. *Nat Neurosci*, 12(5):655–63, May 2009.

12. WI Schöllhorn, G Mayer-Kress, KM Newell, and M Michelbrink. Time scales of adaptive behavior and motor learning in the presence of stochastic perturbations. *Human movement science*, 28(3):319–333, 2009.

13. Ronald J Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning*, 8(3-4):229–256, 1992.

14. Frank Rosenblatt. The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological review*, 65(6):386, 1958.

15. William T Newsome and Edmond B Pare. A selective impairment of motion perception following lesions of the middle temporal visual area (mt). *Journal of Neuroscience*, 8(6):2201–2211, 1988.

16. Kenneth H Britten, Michael N Shadlen, William T Newsome, and J Anthony Movshon. The analysis of visual motion: a comparison of neuronal and psychophysical performance. *Journal of Neuroscience*, 12(12):4745–4765, 1992.

17. M A García-Pérez. Forced-choice staircases with fixed step sizes: asymptotic and small-sample properties. *Vision Res*, 38(12):1861–81, Jun 1998.

18. Yoav Freund and Robert E Schapire. Large margin classification using the perceptron algorithm. *Machine learning*, 37(3):277–296, 1999.

19. Michael Collins. Discriminative training methods for hidden markov models: Theory and experiments with perceptron algorithms. In *Proceedings of the ACL-02 conference on Empirical methods in natural language processing-Volume 10*, pages 1–8. Association for Computational Linguistics, 2002.

20. David JC MacKay. *Information theory, inference and learning algorithms*. Cambridge university press, 2003.

21. Geoffrey E Hinton, Terrence J Sejnowski, and David H Ackley. *Boltzmann machines: Constraint satisfaction networks that learn*. Carnegie-Mellon University, Department of Computer Science Pittsburgh, PA, 1984.

22. Herbert Jaeger. The “echo state” approach to analysing and training recurrent neural networks-with an erratum note. *Bonn, Germany: German National Research Center for Information Technology GMD Technical Report*, 148(34):13, 2001.

23. Wolfgang Maass, Thomas Natschläger, and Henry Markram. Real-time computing without stable states: A new framework for neural computation based on perturbations. *Neural computation*, 14(11):2531–2560, 2002.

24. S. M. McClure. private communication.

25. Janet Metcalfe and Nate Kornell. A region of proximal learning model of study time allocation. *Journal of memory and language*, 52(4):463–477, 2005.

26. Robert A Bjork. Memory and metamemory considerations in the. *Metacognition: Knowing about knowing*, page 185, 1994.

27. Wolfgang Schnotz and Christian Kürschner. A reconsideration of cognitive load theory. *Educational psychology review*, 19(4):469–508, 2007.

28. Jean Piaget and Margaret Cook. *The origins of intelligence in children*, volume 8. International Universities Press New York, 1952.

29. Lev Semenovich Vygotsky. *The collected works of LS Vygotsky: Problems of the theory and history of psychology*, volume 3. Springer Science & Business Media, 1997.

30. Janet Metcalfe. Learning from errors. *Annual review of psychology*, 68:465–489, 2017.

31. Amitai Shenhav, Matthew M Botvinick, and Jonathan D Cohen. The expected value of control: an integrative theory of anterior cingulate cortex function. *Neuron*, 79(2):217–240, 2013.

32. Amitai Shenhav, Sebastian Musslick, Falk Lieder, Wouter Kool, Thomas L Griffiths, Jonathan D Cohen, and Matthew M Botvinick. Toward a rational and mechanistic account of mental effort. *Annual Review of Neuroscience*, (0), 2017.

33. Farran Briggs, George R Mangun, and W Martin Usrey. Attention enhances synaptic efficacy and the signal-to-noise ratio in neural circuits. *Nature*, 499(7459):476–480, 2013.

34. Joshua W Brown and Todd S Braver. Learned predictions of error likelihood in the anterior cingulate cortex. *Science*, 307(5712):1118–1121, 2005.

35. Mihaly Csikszentmihalyi. *Beyond boredom and anxiety.* Jossey-Bass, 2000.

36. Matti Vuorre and Janet Metcalfe. The relation between the sense of agency and the experience of flow. *Consciousness and cognition*, 43:133–142, 2016.

37. Robert Bauer, Meike Fels, Vladislav Royter, Valerio Raco, and Alireza Gharabaghi. Closed-loop adaptation of neurofeedback based on mental effort facilitates reinforcement learning of brain self-regulation. *Clinical Neurophysiology*, 127(9):3156–3164, 2016.

38. W. Kinzel and P. Rujan. Improving a network generalization ability by selecting examples. *Europhysics Letters*, 13(5):473–477, 1990.