Statistical Physics, Bayesian Inference, and Neural Information Processing

Sara A. Solla

Northwestern University

Statistical Physics of Machine Learning
Summer School
Les Houches, July 4-29, 2022

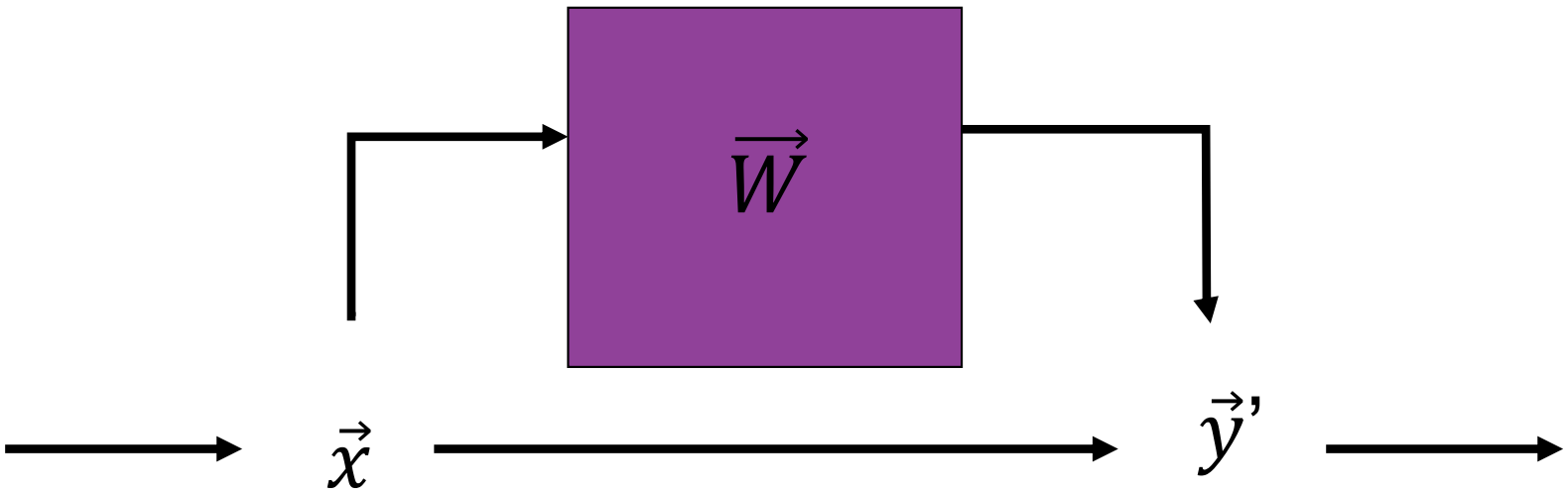
What Does the Brain Do?

Interpret and change the world!

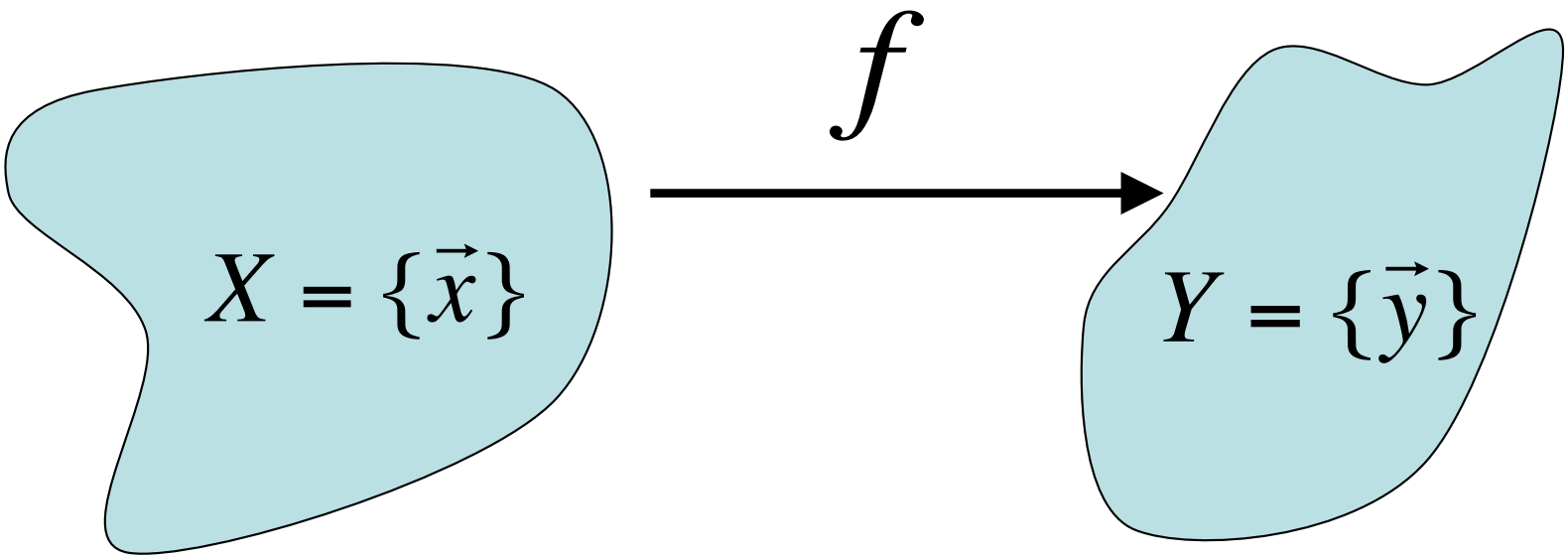
In the world, dynamics and causality:

$$\vec{x} \longrightarrow \vec{y}$$

The brain receives the same input, processes it, and affects the output:



Input-output maps

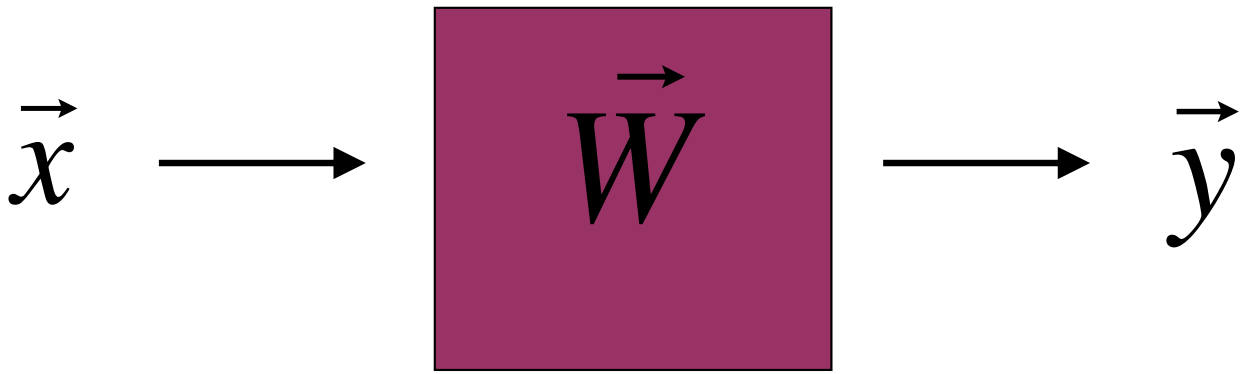


$$\vec{x} = \{x_1, x_2, \ldots, x_N\} \rightarrow \vec{y} = \{y_1, y_2, \ldots, y_R\}$$

$$\vec{y} = f(\vec{x})$$

Input-output maps

$$\vec{y} = f_{\vec{W}}(\vec{x})$$

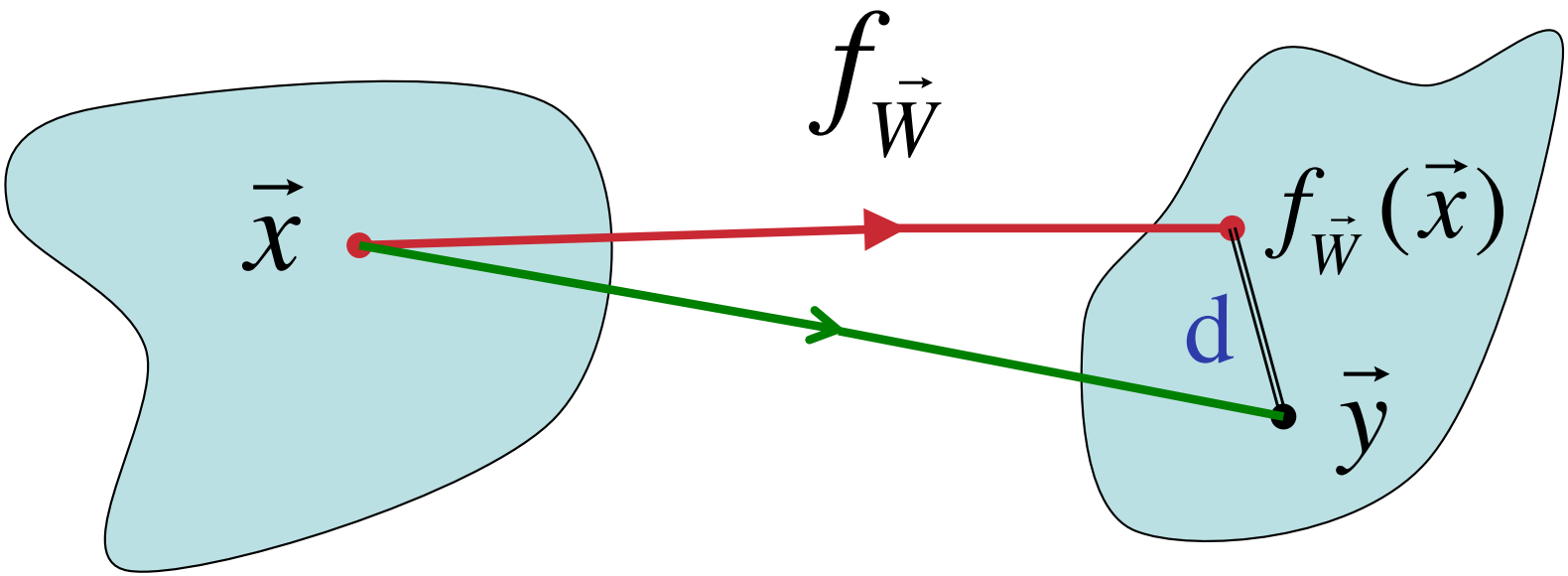


What specifies the value of the parameters $\vec{W}$?

Data: $\vec{\xi}^{\mu} = (\vec{x}^{\mu}, \vec{y}^{\mu})$ $\quad 1 \le \mu \le m$

Examples of the desired map: m input-output pairs

Learning from examples



Given an example of the desired map, the error made by a specific module $\vec{W}$ on this example is:

$$E(\vec{W} \,|\, \vec{x}, \vec{y}) = d\left(\vec{y}, f_{\vec{W}}(\vec{x})\right) = \tfrac{1}{2}\left(\vec{y} - f_{\vec{W}}(\vec{x})\right)^2$$

Learning error

Given a training set of size m:

$$\vec{\xi}^{\mu} = (\vec{x}^{\mu}, \vec{y}^{\mu}) \qquad 1 \le \mu \le m$$

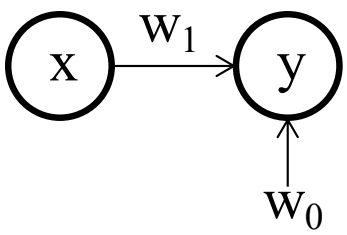
construct a cost function that measures the average error over the training set, the learning error:

$$E_L(\vec{W}) = (1/m) \sum_{\mu=1}^{m} E(\vec{W} \left| \vec{x}^{\mu}, \vec{y}^{\mu} \right.)$$

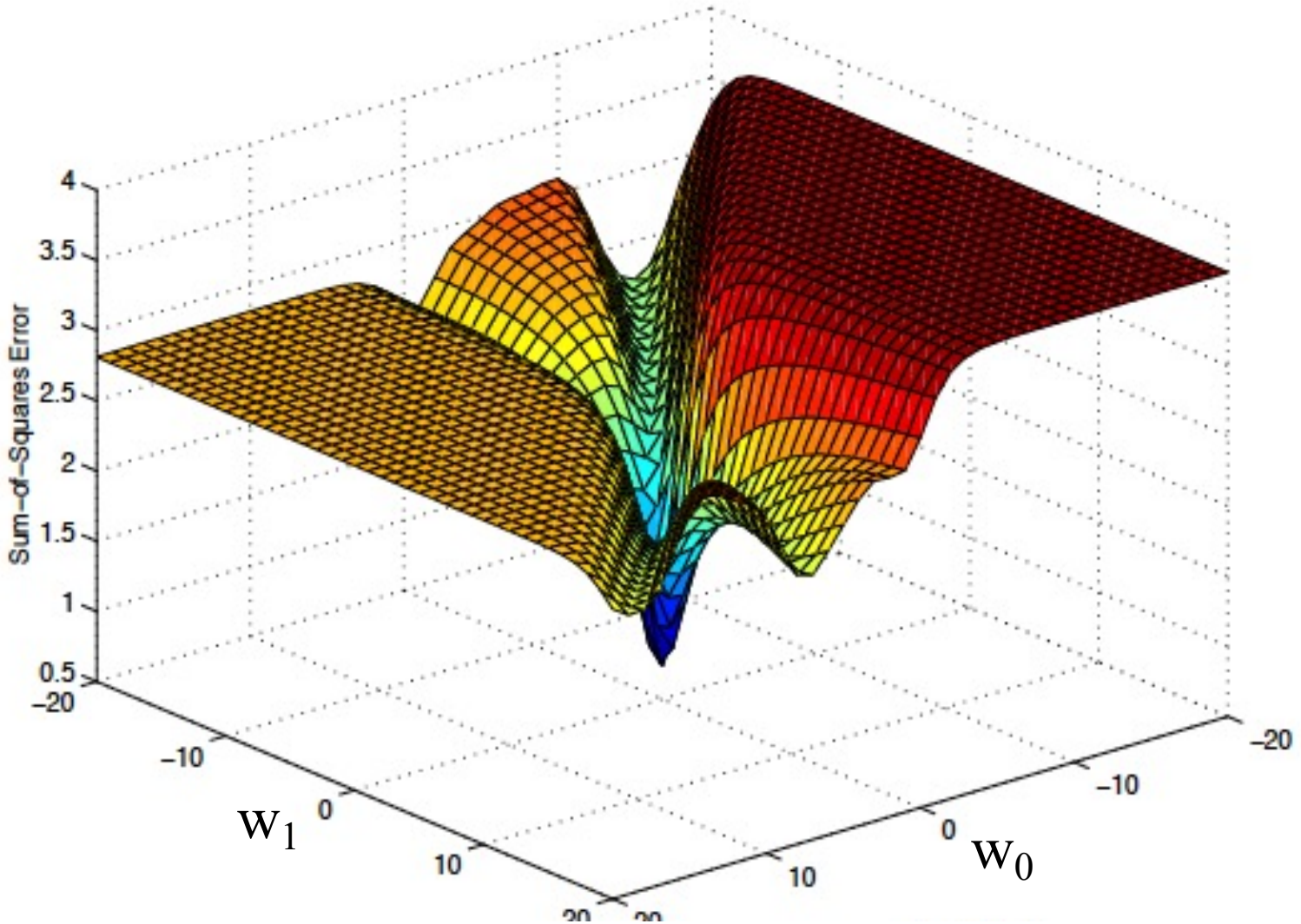
Most learning algorithms are based on finding the parameters $\vec{W}^*$ that minimize this learning error.

Learning by gradient descent

Learning error



$$y = g\left(w_1 x + w_0\right)$$



Learning by gradient descent

Perceptron learning by gradient descent

$$y = g\left(\sum_{i=1}^{N} w_i\, x_i + w_0\right) = g\left(\vec{w}^T \vec{x}\right)$$

g: soft nonlinearity

Learning from examples: $\vec{\xi}^{\mu} = (\vec{x}^{\mu},\, y^{\mu})$

Error on μ-th example: $$E^{\mu} = \frac{1}{2}\left(y^{\mu} - g\left(\sum_{i=1}^{N} w_i\, x_i^{\mu} + w_0\right)\right)^2$$

Error gradient: $$\frac{\partial E^{\mu}}{\partial w_i} = -\left(y^{\mu} - g\left(\vec{w}^T \vec{x}^{\mu}\right)\right) g'\left(\vec{w}^T \vec{x}^{\mu}\right) x_i^{\mu}$$

Gradient descent learning: delta rule

$$\frac{\partial E^{\mu}}{\partial w_i} = -\left(y^{\mu} - g\left(\vec{w}^T \vec{x}^{\mu}\right)\right) g'\left(\vec{w}^T \vec{x}^{\mu}\right) x_i^{\mu} \equiv -\delta^{\mu}\, x_i^{\mu}$$

$$\delta^{\mu} \equiv g'\left(\vec{w}^T \vec{x}^{\mu}\right)\left(y^{\mu} - g\left(\vec{w}^T \vec{x}^{\mu}\right)\right)$$

$$w_i \rightarrow w_i + \Delta w_i = w_i - \eta \frac{\partial E^{\mu}}{\partial w_i} = w_i + \eta\, \delta^{\mu}\, x_i^{\mu}$$

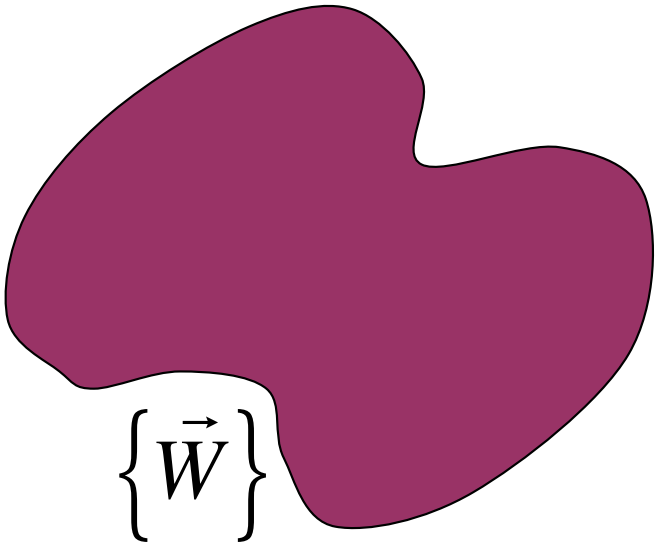
$$\Delta w_i^{\mu} = \eta\, \delta^{\mu}\, x_i^{\mu}$$

Configuration Space

For each example $\vec{\xi}^{\mu} = (\vec{x}^{\mu}, \vec{y}^{\mu})$ in the training set, define a masking function:

$$\Theta(\vec{W}, \vec{\xi}^{\mu}) = 1 \quad \text{if} \quad f_{\vec{W}}(\vec{x}^{\mu}) = \vec{y}^{\mu}$$

$$\Theta(\vec{W}, \vec{\xi}^{\mu}) = 0 \quad \text{if} \quad f_{\vec{W}}(\vec{x}^{\mu}) \neq \vec{y}^{\mu}$$

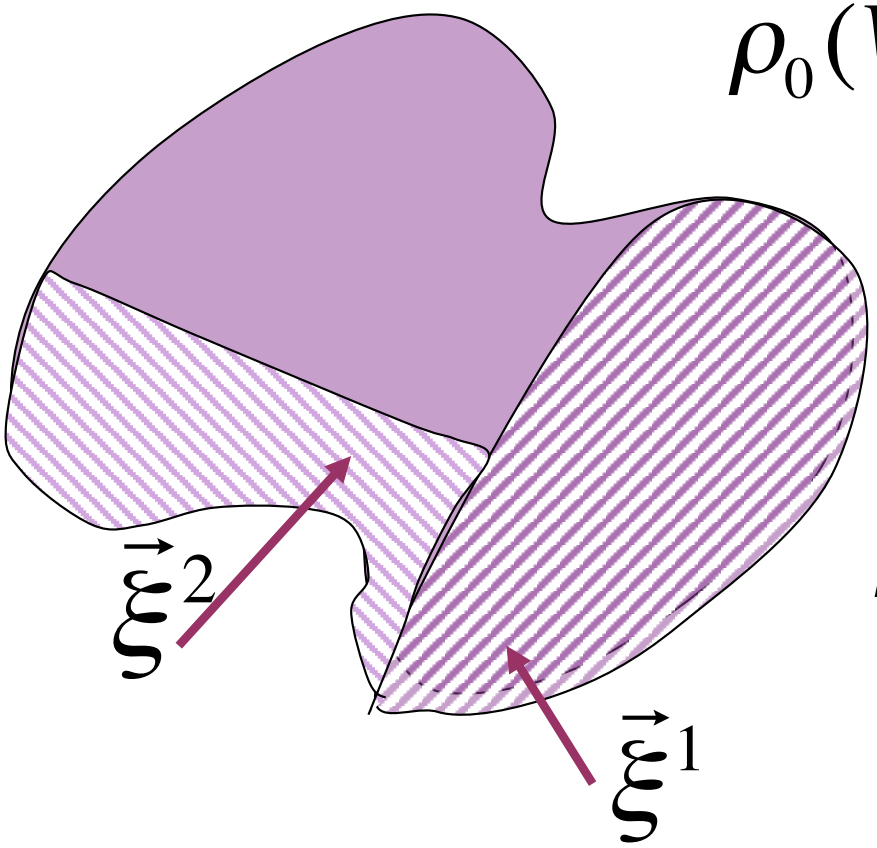


Prior $\rho_0(\vec{W})$

Normalization:

$$\int \rho_0(\vec{W})\, d\vec{W} = 1$$

Error-Free Learning



$$\rho_0(\vec{W}) \Rightarrow$$

$$\rho_0(\vec{W})\Theta(\vec{W},\vec{\xi}^{\,1}) \Rightarrow$$

$$\rho_0(\vec{W})\Theta(\vec{W},\vec{\xi}^{\,1})\Theta(\vec{W},\vec{\xi}^{\,2})$$

Masking: $$Z_m = \int d\vec{W}\,\rho_0(\vec{W})\prod_{\mu=1}^{m}\Theta(\vec{W},\vec{\xi}^{\,\mu})$$

Contraction: $$Z_m \le Z_{m-1} \le \ldots \le Z_1 \le Z_0 = 1$$

Learning from Noisy Data

Consider the error on the μth example:

$$E(\vec{W}\left|\vec{\xi}^{\mu}\right.) = d\left(\vec{y}^{\mu}, f_{\vec{W}}(\vec{x}^{\mu})\right)$$

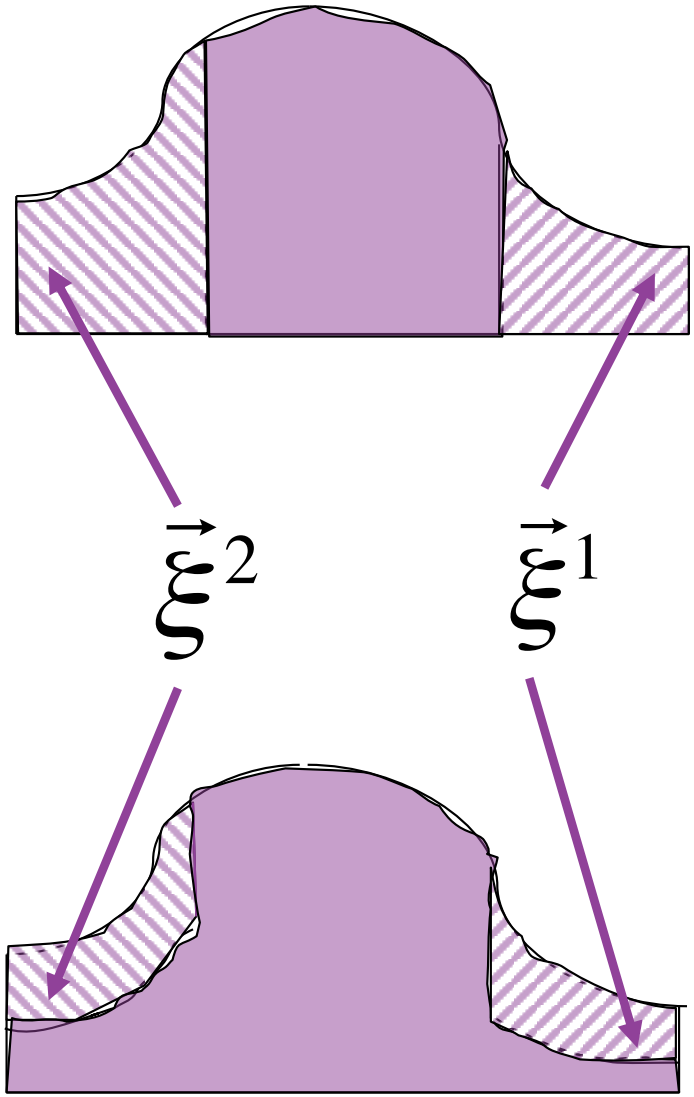
$$\text{If } f_{\vec{W}}(\vec{x}^{\mu}) = \vec{y}^{\mu}, E(W\left|\vec{\xi}^{\mu}\right.) = 0 \Rightarrow \Theta(\vec{W}, \vec{\xi}^{\mu}) = 1$$

$$\text{If } f_{\vec{W}}(\vec{x}^{\mu}) \neq \vec{y}^{\mu}, \text{ instead of setting } \Theta(\vec{W}, \vec{\xi}^{\mu}) = 0$$

introduce a survival probability:

$$\Theta(\vec{W}, \vec{\xi}^{\mu}) \rightarrow \exp\left(-\beta E(\vec{W}\left|\vec{\xi}^{\mu}\right.)\right)$$

Hard vs Soft Masking



Hard masking: configurations incompatible with the data are eliminated.

Soft masking: configurations are attenuated by a factor exponentially controlled by the error made on the data.

Learning with Uncertainty

$$\rho_0(\vec{W}) \;\Rightarrow\; \rho_0(\vec{W})\exp\left(-\beta E(\vec{W}\left|\vec{\xi}^{\,1}\right.)\right) \;\Rightarrow$$

$$\rho_0(\vec{W})\exp\left(-\beta E(\vec{W}\left|\vec{\xi}^{\,1}\right.)\right)\exp\left(-\beta E(\vec{W}\left|\vec{\xi}^{\,2}\right.)\right)$$

$$Z_m = \int d\vec{W}\,\rho_0(\vec{W})\prod_{\mu=1}^{m}\exp\left(-\beta E(\vec{W}\left|\vec{\xi}^{\,\mu}\right.)\right)$$

$$Z_m = \int d\vec{W}\,\rho_0(\vec{W})\exp\left(-m\beta E_L(\vec{W})\right)$$

with learning error: $E_L(\vec{W}) = (1/m)\sum_{\mu=1}^{m} E(\vec{W}\left|\vec{\xi}^{\,\mu}\right.)$

Gibbs Distribution

The ensemble of all possible modules is described by the prior density $\rho_0(\vec{W})$. The ensemble of trained modules is described by the posterior density $\rho_m(\vec{W})$:

$$\rho_m(\vec{W}) = \frac{1}{Z_m}\rho_0(\vec{W})\exp\left(-\beta m E_L(\vec{W})\right)$$

Note that $\int d\vec{W}\,\rho_m(\vec{W}) = 1$, and that the partition function Z_m provides the normalization constant. Note also that this distribution arises from without invoking specific algorithms for exploring the configuration space $\{\vec{W}\}$.

Natural Statistics

Training data $\vec{\xi} = (\vec{x}, \vec{y})$ is drawn from a distribution $\tilde{P}(\vec{\xi}) = \tilde{P}(\vec{x}, \vec{y}) = \tilde{P}(\vec{y} \,|\, \vec{x})\, \tilde{P}(\vec{x})$

$\tilde{P}(\vec{x})$ describes the region of interest input space

$\tilde{P}(\vec{y} \,|\, \vec{x})$ describes the functional dependence

Thermodynamics of Learning

The partition function

$$Z_m = \int d\vec{W}\, \rho_0(\vec{W}) \exp\left(-\beta \sum_{\mu=1}^{m} E(\vec{W} \left| \vec{\xi}^{\mu} \right.) \right)$$

depends on the specific set of data points $D = \left\{ \vec{\xi}^{\mu} \right\}$ drawn from $\tilde{P}(\vec{\xi})$. The associated free energy

$$F = -(1/\beta) \left\langle \left\langle \ln Z_m \right\rangle \right\rangle_D$$

follows from averaging over all possible data sets of size m. The average learning error follows from the usual thermodynamic derivative:

$$E_L = -\frac{1}{m} \frac{\partial}{\partial \beta} \left\langle \left\langle \ln Z_m \right\rangle \right\rangle_D$$

Entropy of Learning

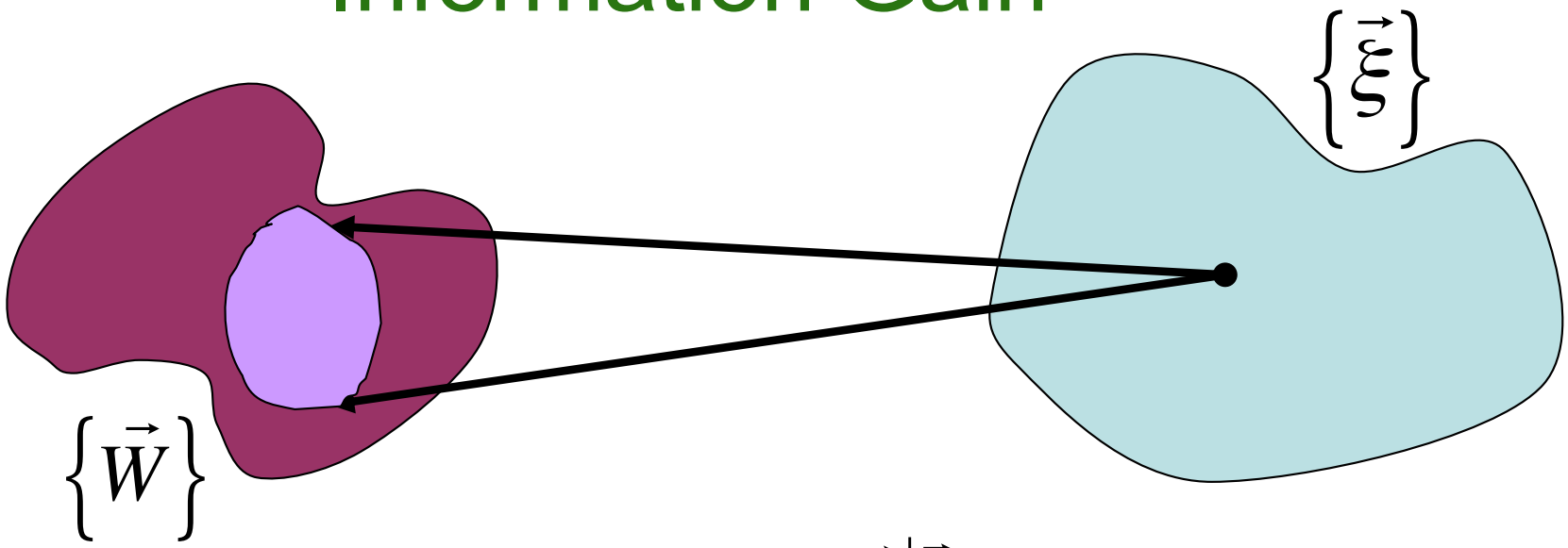
The entropy follows from $F = m E_L - (1/\beta) S$

For the learning process, this results in:

$$S = -\int d\vec{W}\, \rho_m(\vec{W}) \ln\left[\frac{\rho_m(\vec{W})}{\rho_0(\vec{W})}\right] = -D_{KL}\left[\rho_m | \rho_0\right]$$

The entropy of learning is minus the Kullback-Leibler distance between the posterior $\rho_m(\vec{W})$ and the prior $\rho_0(\vec{W})$, and it measures the amount of information gained. The distance between posterior and prior increases monotonically with the size m of the training set.

Information Gain



$P(\vec{W}) = \rho_0(\vec{W})$: prior distribution

$P(\vec{W}|\vec{\xi})$: distribution induced by example $\vec{\xi}$

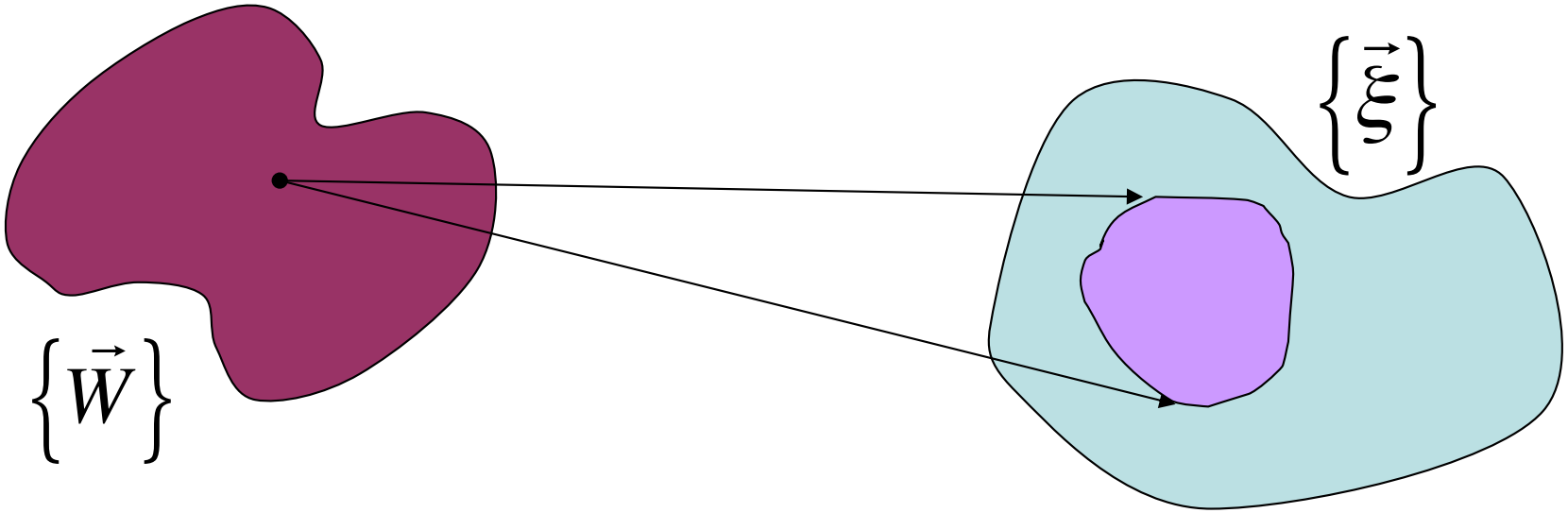
The entropy difference $\Delta H = H_{P(\vec{W})} - \left\langle \left\langle H_{P(\vec{W}|\vec{\xi})} \right\rangle \right\rangle_{P(\vec{\xi})}$

can be shown to be equal to the mutual information between the $\{\vec{W}\}$ space and the $\{\vec{\xi}\}$ space.

the brain

the world

Maximum Likelihood Learning



$P(\vec{\xi}|\vec{W})$: distribution induced through hypothesis $\vec{W}$

$\tilde{P}(\vec{\xi})$: true distribution

Likelihood of the data:

$$\mathcal{L}(\vec{W}) = P(D|\vec{W}) = P(\vec{\xi}^1, \vec{\xi}^2, \ldots, \vec{\xi}^m|\vec{W}) = \prod_{\mu=1}^{m} P(\vec{\xi}^\mu|\vec{W})$$

BUT: what is the form of $P(\vec{\xi}|\vec{W})$?

Learning Coherence

Two approaches to learning:

- Minimize the error on the data:

$$E_L(\vec{W}) = \sum_{\mu=1}^{m} E(\vec{W}\left|\vec{\xi}^{\mu}\right.)$$

- Maximize the likelihood of the data:

$$\mathcal{L}(\vec{W}) = \prod_{\mu=1}^{m} P(\vec{\xi}^{\mu}\left|\vec{W}\right.)$$

Require that these two approaches be coherent!

$$P(\vec{\xi}\left|\vec{W}\right.) = \frac{1}{z(\beta)} \exp\left(-\beta\, E(\vec{W}\left|\vec{\xi}\right.)\right)$$

(Appendix)

Bayesian Learning

We now compute the likelihood of the data: $P(D|\vec{W}) =$

$$\prod_{\mu=1}^{m} P(\vec{\xi}^{\mu}|\vec{W}) = \frac{1}{z(\beta)^m} \exp\left(-\beta \sum_{\mu=1}^{m} E(\vec{\xi}^{\mu}|\vec{W})\right) = \frac{1}{z(\beta)^m} \exp\left(-\beta m E_L(\vec{W})\right)$$

Bayesian inversion:

$$P(\vec{W}|D) = \frac{P(D|\vec{W}) * P(\vec{W})}{P(D)}$$

Gibbs distribution:

$$\rho_m(\vec{W}) = \frac{1}{Z_m} \rho_0(\vec{W}) \exp\left(-\beta m E_L(\vec{W})\right)$$

Bayes ⟺ Gibbs

Prior: $$P(\vec{W}) \Leftrightarrow \rho_0(\vec{W})$$

Posterior: $$P(\vec{W}|D) \Leftrightarrow \rho_m(\vec{W})$$

Likelihood: $$P(D|\vec{W}) \Leftrightarrow \frac{1}{z(\beta)^m} \exp\left(-\beta m E_L(\vec{W})\right)$$

Evidence: $$P(D) \Leftrightarrow \frac{1}{z(\beta)^m} Z_m$$

where $$P(D) = \int d\vec{W}\, P(D|\vec{W}) P(\vec{W})$$

The normalization constant $z(\beta)$ plays a role in the evaluation of prediction errors (has the brain acquired a good model of the world?)

Generalization Ability

Consider a new point $\vec{\xi}$ not part of the training data $D = \{\vec{\xi}^1, \vec{\xi}^2, ..., \vec{\xi}^m\}$. What is the likelihood of this test point?

$$P(\vec{\xi}|D) = \int d\vec{W}\, P(\vec{\xi}|\vec{W}) P(\vec{W}|D)$$

with: $$P(\vec{\xi}|\vec{W}) = \frac{1}{z(\beta)} \exp\left(-\beta E(\vec{W}|\vec{\xi})\right)$$

and: $$P(\vec{W}|D) = \rho_m(\vec{W}) = \frac{1}{Z_m} \rho_0(\vec{W}) \exp\left(-\beta \sum_{\mu=1}^{m} E(\vec{W}|\vec{\xi}^{\mu})\right)$$

Generalization Ability

$$P(\vec{\xi}|D) = \int d\vec{W}\, P(\vec{\xi}|\vec{W})P(\vec{W}|D) =$$

$$= \frac{1}{z(\beta)Z_m} \int d\vec{W}\, \rho_0(\vec{W}) \exp\left(-\beta \sum_{\mu=1}^{m+1} E(\vec{W}|\vec{\xi}^{\mu}) \right)$$

Where $\vec{\xi}^{m+1} = \vec{\xi}$: the test point appears as if it had been added to the training set. Thus:

$$P(\vec{\xi}|D) = \frac{Z_{m+1}}{z(\beta)Z_m}$$

Generalization Error

The generalization error is defined through the ln of the likelihood of the test point $\vec{\xi}$:

$$P(\vec{\xi}|D) = \frac{Z_{m+1}}{z(\beta)Z_m} \quad \Rightarrow \quad E_G = -\frac{1}{\beta}\left[\ln\frac{Z_{m+1}}{Z_m} - \ln z(\beta)\right]$$

For large m, the difference between $(\ln Z_{m+1})$ and $(\ln Z_m)$ can be approximated by a derivative with respect to m. Then $(\ln Z)$ is averaged over all possible data sets of size m, to obtain:

$$E_G = -\frac{1}{\beta}\frac{\partial}{\partial m}\left\langle\left\langle \ln Z_m \right\rangle\right\rangle_D + \frac{1}{\beta}\ln z(\beta)$$

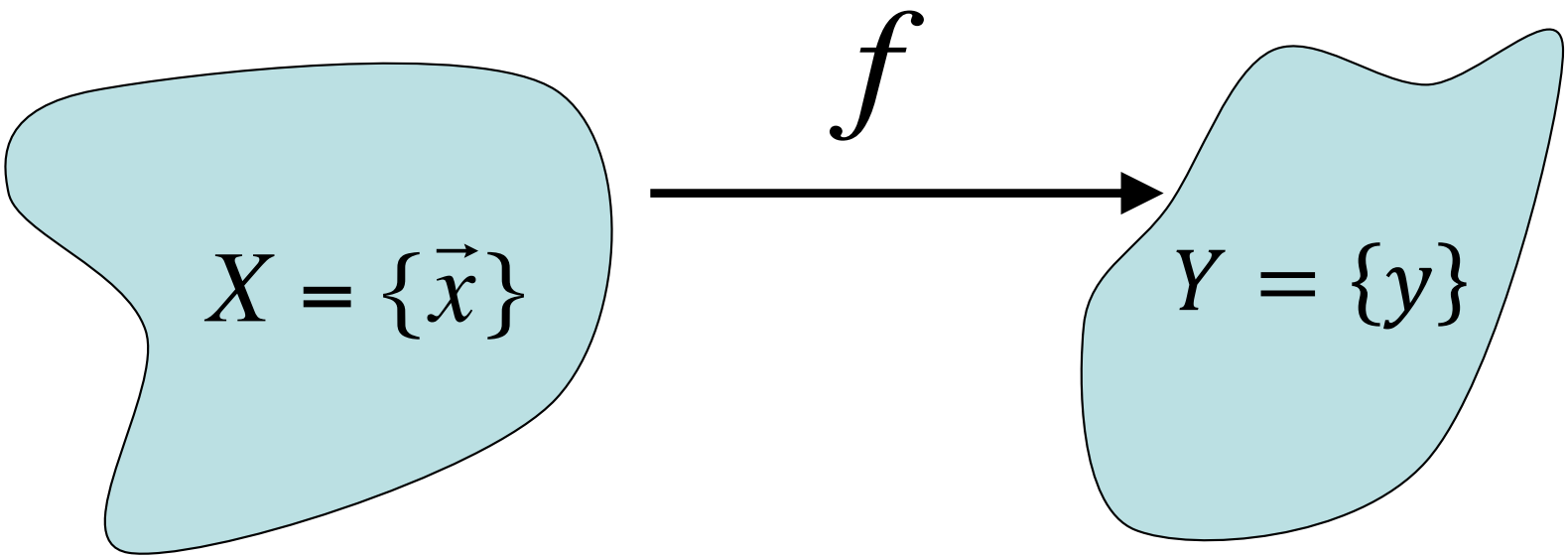
Learning vs Generalization

Two thermodynamic derivatives:

$$E_L = -\frac{1}{m}\frac{\partial}{\partial \beta}\left\langle\left\langle \ln Z_m \right\rangle\right\rangle_D$$

$$E_G = -\frac{1}{\beta}\frac{\partial}{\partial m}\left\langle\left\langle \ln Z_m \right\rangle\right\rangle_D + \frac{1}{\beta}\ln z(\beta)$$

A simple example: linear map



$$\vec{x} = \{x_1, x_2, \ldots, x_N\} \rightarrow y = f_W(\vec{x}) = \sum_{i=1}^{N} W_i \; x_i = \vec{W}^T \; \vec{x}$$

$\{\vec{W}\}$ drawn from $\rho_0(\vec{W}) = \mathcal{N}(0, C_w)$

with $C_w = \sigma_w^2 I_N$ and $\sigma_w \gg 1$

A simple example: examples

$$\tilde{P}(\vec{x}) = \mathcal{N}(0, C_x) \text{ with } C_x = \sigma_x^2 I_N$$

Target output for input $\vec{x}^\mu$ is

$$y^\mu = \vec{W}_0^T \vec{x}^\mu + \eta^\mu$$

$$\tilde{P}(y|\vec{x}) = \mathcal{N}\left(\vec{W}_0^T \vec{x}^\mu, \sigma_\eta^2\right)$$

Train to minimize $E_L\left(\vec{W}\right) = \frac{1}{2}\sum_{\mu=1}^{m}\left(y^\mu - \vec{W}^T\vec{x}^\mu\right)^2 =$

$$= \frac{1}{2}\sum_{\mu=1}^{m}\left(\left(\vec{W} - \vec{W}_0\right)^T\vec{x}^\mu - \eta^\mu\right)^2$$

A simple example: partition function

For $\sigma_w \gg 1$ and in the large m limit the free energy can be computed analytically:

$$\langle\langle \ln Z_m \rangle\rangle = -N \ln \sigma_w - \frac{\mathbf{W}_0^T \cdot \mathbf{W}_0}{2\sigma_w^2} - \frac{N}{2} \ln(2\beta\sigma_x^2 m)$$
$$+ (N-m)\beta\sigma_\eta^2 + O(1/m)$$

The thermodynamic derivatives are:

$$E_L = \frac{N}{2m\beta} + \left(1 - \frac{N}{m}\right)\sigma_\eta^2 + O(1/m^2)$$

$$E_G = \frac{N}{2m} + \beta\sigma_\eta^2 + \ln\left(\frac{\pi}{\beta}\right)^{1/2} + O(1/m^2)$$

A simple example: effective temperature

Effective temperature β_0 associated to the noise in the examples:

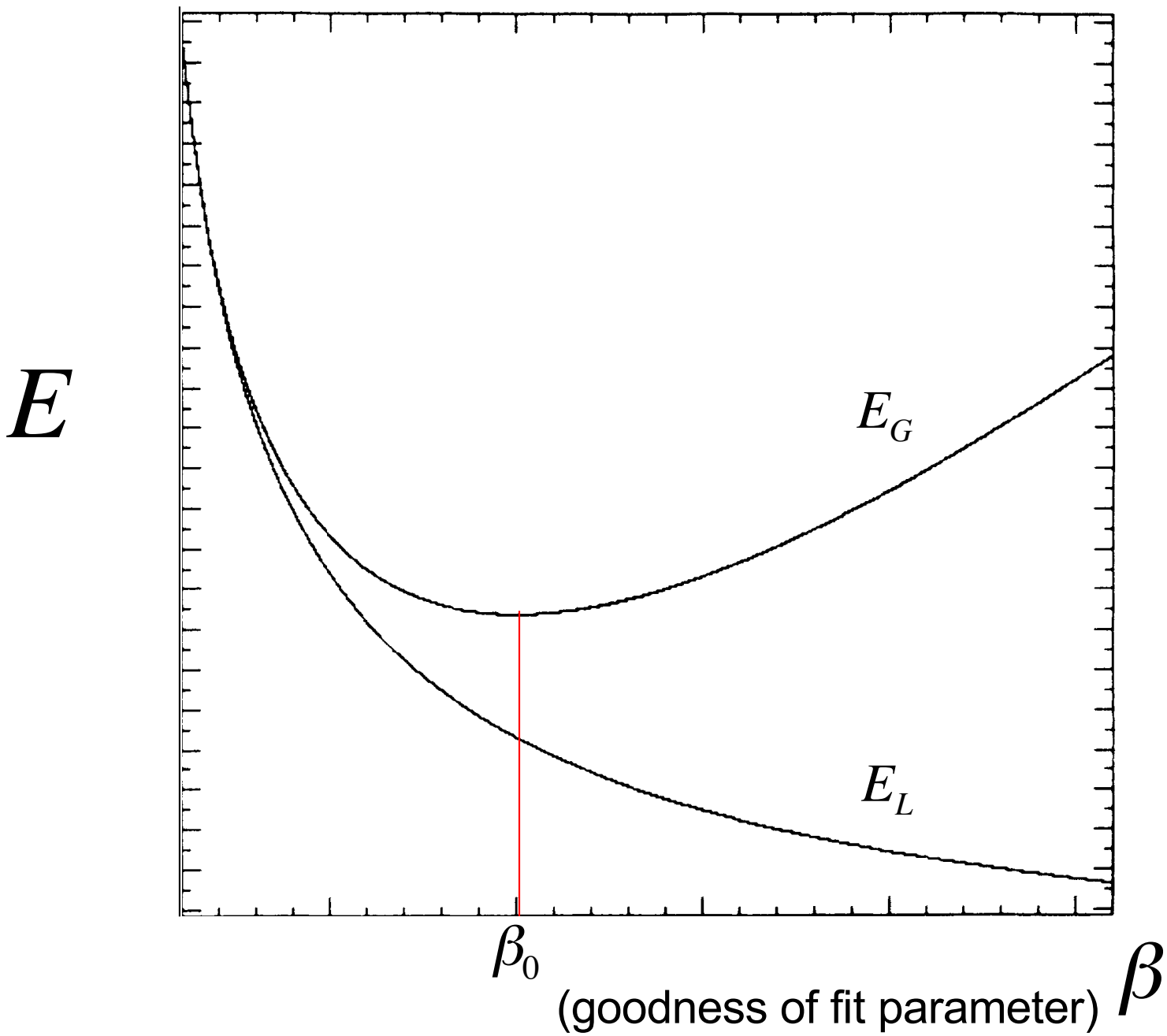
$$\beta_0 = \frac{1}{2\sigma_\eta^2}$$

The thermodynamic derivatives are:

$$E_L = \frac{N}{2m}\left[\frac{1}{\beta} - \frac{1}{\beta_0}\right] + \frac{1}{2\beta_0} + O(1/m^2)$$

$$E_G = \frac{N}{2m} + \frac{\beta}{2\beta_0} + \ln\left[\frac{\pi}{\beta}\right]^{1/2} + O(1/m^2)$$

Learning vs generalization



Appendix.1

Require that the minimization of the learning error:

$$E_L(\vec{W}) = \sum_{\mu=1}^{m} E(\vec{W}\left|\vec{\xi}^{\mu}\right.)$$

guarantees the maximization of the likelihood:

$$\mathcal{L}(\vec{W}) = \prod_{\mu=1}^{m} P(\vec{\xi}^{\mu}\left|\vec{W}\right.)$$

Given a training set $\left(\vec{\xi}^{1}, \vec{\xi}^{2}, ..., \vec{\xi}^{m}\right)$, these two functions need to be related:

$$\mathcal{L}\left(\vec{W}\right) = \Phi\left(E_L\left(\vec{W}\right)\right)$$

Appendix.2

Take a derivative on both sides with respect to one of the points in the training set, $\vec{\xi}_j$:

$$\frac{\partial \mathcal{L}\left(D\middle|\vec{W}\right)}{\partial \vec{\xi}_j} = \mathcal{L}\left(D\middle|\vec{W}\right)\frac{1}{P\left(\vec{\xi}_j\middle|\vec{W}\right)}\frac{\partial P\left(\vec{\xi}_j\middle|\vec{W}\right)}{\partial \vec{\xi}_j} =$$

$$= \Phi'\frac{\partial E\left(\vec{W}\middle|\vec{\xi}_j\right)}{\partial \vec{\xi}_j}$$

This leads to:

$$\frac{\Phi'}{\Phi} = \frac{\dfrac{1}{P\left(\vec{\xi}_j\middle|\vec{W}\right)}\dfrac{\partial P\left(\vec{\xi}_j\middle|\vec{W}\right)}{\partial \vec{\xi}_j}}{\dfrac{\partial E\left(\vec{W}\middle|\vec{\xi}_j\right)}{\partial \vec{\xi}_j}}$$

Appendix.3

While the left-hand side of the equation depends on the full training set $\left(\vec{\xi}^1, \vec{\xi}^2, ..., \vec{\xi}^m\right)$, the right-hand side depends only on $\vec{\xi}^j$. The only way for this equality to hold for all values of $\left(\vec{\xi}^1, \vec{\xi}^2, ..., \vec{\xi}^m\right)$ is for both sides to be actually independent of the data, and thus equal to a constant:

$$\frac{\frac{1}{P\left(\vec{\xi}_j \middle| \vec{W}\right)} \frac{\partial P\left(\vec{\xi}_j \middle| \vec{W}\right)}{\partial \vec{\xi}_j}}{\frac{\partial E\left(\vec{W} \middle| \vec{\xi}_j\right)}{\partial \vec{\xi}_j}} = -\beta$$

Appendix.4

The equation $$\frac{1}{P\left(\vec{\xi}_j\middle|\vec{W}\right)}\frac{\partial P\left(\vec{\xi}_j\middle|\vec{W}\right)}{\partial\vec{\xi}_j} = -\beta\frac{\partial E\left(\vec{W}\middle|\vec{\xi}_j\right)}{\partial\vec{\xi}_j}$$

leads to $$P\left(\vec{\xi}_j\middle|\vec{W}\right)\propto\exp\left(-\beta E\left(\vec{W}\middle|\vec{\xi}_j\right)\right)$$

The normalized probability distribution is:

$$P\left(\vec{\xi}\middle|\vec{W}\right)=\frac{1}{z(\beta)}\exp\left(-\beta E\left(\vec{W}\middle|\vec{\xi}\right)\right)$$

with $$z(\beta)=\int d\vec{\xi}\exp\left(-\beta E\left(\vec{W}\middle|\vec{\xi}\right)\right)$$

Since the equation that determines $P\left(\vec{\xi}\middle|\vec{W}\right)$ is first order, there is only one constant of integration: β. For $\beta>0$, minima of E correspond to maxima of P.