For a manifold M with a metric g, we can define $\hat{g} : TM \to TM^*$ a map between (homomorphism) tangent bundles.

$$\hat{g}(X)(Y) = g(X, Y)$$

$X, Y \in TM$ are vector fields

Then the gradient can be defined as the vector field s.t. $\langle \text{grad } f, X \rangle_g = Xf$,

$(\forall) f \in C^\infty(M), X \in TM$.

$\text{grad } f \in TM$.

From here, you can see that $\text{grad } f = \hat{g}^{-1}(df)$.

So, for our calculations $g^{-1} \nabla_\theta f$ will be the "new gradient". g^{-1} here is a matrix and θ are the parameters of the model.

We can see that when updating the parameters,

$\theta_{t+1} = \theta_t - \eta B \nabla_\theta L_1$ for some loss function L, learning rate η and the preconditioner B, we can add this B controlling the connections between gradients. If we write a second order approximation, we get the Hessian which is very inefficient to compute. Therefore, we have to construct a preconditioner B which is more efficient to compute and can recover some of the properties of the Hessian, namely it tells us about the rate of change of gradients. The candidate is the Fisher matrix, the second derivative of the KL:

$$F(\theta) = \mathbb{E}_{x,y \sim p(x,y;\theta)}\left[\nabla_\theta \nabla_\theta^T\right],$$

where $\nabla_\theta := -\nabla_\theta \log p(x, y; \theta)$.

$p(x, y_i; \theta)$ is as usual a joint over the data x, the target y_i, depending on parameters θ.

The first thing is to get rid of integrals, so we use Monte Carlo approximations:

$$F_{approx.}(\theta) = \frac{1}{\bar{m}} \sum_{i=1}^{\bar{m}} \nabla_{\theta,i} \nabla_{\theta,i}^T$$

where $\bar{m}$ is the number of labeled training samples.

Using the notation $B_i(\theta) = \frac{1}{\bar{m}} \nabla_{\theta,i} \nabla_{\theta,i}^T$

$$\nabla_{\theta,i} = -\nabla_\theta \log p(y_i | x, \theta)$$

$B_i(\theta)$ can be moreover approximated to be diagonal (without cross-layer terms).

By the KFAC approximation, if $B_i(\theta) = \text{diag}(B_{1,i}, \ldots, B_{m,i})$, then $B_{k,i} = (U_{k,i} \otimes$

$V_{k,i}) = U_{k,i} \otimes V_{k,i}$, where $U_{k,i}$ and $V_{k,i}$ are smaller matrices. Then, the preconditioned gradient $B_{k,i} \nabla_{\theta_{k,i}} = U_{k,i}^{-1} \frac{\partial l}{\partial W_k} V_{k,i}^{-1}$.

$$\frac{\partial l}{\partial W_k} = \frac{\partial l}{\partial X_k} \odot \phi_k'(W_k \tilde{X}_{k-1}) \tilde{X}_{k-1}^T$$

$\odot$: point-wise

$$= u_{k,i} v_{k,i}^T$$

So, U_k, V_k can be approximated as the expected values of $u_{k,i} u_{k,i}^T$, $v_{k,i} v_{k,i}^T$.

Hence, $U_k = \frac{1}{n}\left(\frac{\partial l}{\partial X_k} \odot \phi_k'(W_k \tilde{X}_{k-1})\right)$

$$\left(\frac{\partial l}{\partial X_k} \odot \phi_k'(W_k \tilde{X}_{k-1})\right)^T$$

$$V_k = \frac{1}{n} \tilde{X}_{k-1} \tilde{X}_{k-1}^T$$