Appendix A | Measuring parameter contribution

Marginalization of parameters

In models that are able to handle the lack of information about a particular representation like in natural Bayesian classifiers, the contribution can be measured by marking the representation as unknown. Typically though, neural networks are not able to handle missing information and setting the parameters of a representation to zero will still result in interpretable information for subsequent layers. While removing a feature representation and retraining the network would alleviate this issue, quantifying the contribution of thousands of representations this way is generally infeasible. Instead we make use of the models classification probabilities given by the softmax activation output which allows us to estimate the classification probability while lacking a representation by marginalizing it out via standard method from statistics. Marginalization effectively computes the weighted average of the classification probabilities after the representation has been replaced with random values sampled from an appropriate distribution. See equation A.1 for the mathematical definition used for our evaluation.

$$p(y|x, \Theta_{\setminus\theta}) = \sum_{\theta} p(y|x, \Theta)p(\theta) \tag{A.1}$$

$p(y|x, \Theta)$ defines here the probability of input x belonging to class y and $p(y|x, \Theta_{\setminus\theta})$ the probability if θ is unknown. Note that a feature representation is represented by its parameters θ, which in turn consists classically of a weight w and a potential bias b in a neural network setting. Θ defines then the set of all parameters such that $\theta \in \Theta$. Each classification probability is eventually weighted by the prior probability of the sample θ expressing the likelihood the parameter in question takes value θ. We used 100 samples in our experiments to approximate the contribution.

Derivation

Given a parametric model like a DNN that is described by its parameters Θ, we can express the probability of input x belonging to class y as $p(y|x, \Theta)$, where the probabilities are given by the softmax output layer. To measure the contribution of a feature generated by parameter $\theta \in \Theta$, we are interested in what the probability is when θ is missing or unknown. By assuming that the input is independent of the parameters as well as the parameters are independent of each other, such that $p(x, \Theta) = p(x)p(\Theta)$ and $p(\Theta) = p(\Theta_{\setminus\theta})p(\theta)$ and by treating the parameters as random variables we can marginalize out θ as follows.

$$
\begin{aligned}
p(y|x,\Theta_{\backslash\theta}) &= \frac{\int_\theta p(y,x,\Theta)d\theta}{\int_\theta p(x,\Theta)d\theta} \\
&= \frac{\int_\theta p(y|x,\Theta)p(x,\Theta_{\backslash\theta})p(\theta)d\theta}{p(x,\Theta_{\backslash\theta})\int_\theta p(\theta)d\theta} \\
&= \int_\theta p(y|x,\Theta)p(\theta)d\theta
\end{aligned} \tag{A.2}
$$

As the integral over all possible values of θ is intractable for DNN-like structures, we instead approximate the probability by sampling from θ a finite number of times. We can now express the upper equation with a sum over all samples of θ.

$$p(y|x,\Theta_{\backslash\theta}) = \sum_\theta p(y|x,\Theta)p(\theta) \tag{A.3}$$

To sample from θ, we assume that the values are normal distributed with uniform variance and mean centered at the learned weight w and bias b:

$$\theta \sim \mathcal{N}(\mu = w,\, \Sigma = I), \mathcal{N}(\mu = b,\, \Sigma = I)$$

so that

$$p(\theta) = p_{\mathcal{N}(w,I)}(w) \cdot p_{\mathcal{N}(b,I)}(b) \tag{A.4}$$

Generalizing contributions from classes to tasks

As proposed by (Robnik-Sikonja & Kononenko, 2008), we use the weighted evidence (WE) to measure the contribution of parameter θ towards class probability $p(y|x,\Theta)$ (see equation A.6) instead of taking the difference of probabilities directly. $WE_\theta(y|x,\Theta)$ gives us a positive value indicating θ adds evidence for class y for input x, while a negative value adds evidence against class y and zero if θ has no contribution at all. To eventually determine the contribution towards a class independent of the input we calculate the arithmetic mean of the absolute weighted evidence over more than 500 input samples (see equation A.7) from the test set.

$$odds(z) = \frac{p(z)}{1-p(z)} \tag{A.5}$$

$$WE_\theta(y|x,\Theta) = log_2(odds(y|x,\Theta)) - log_2(odds(y|x,\Theta_{\backslash\theta})) \tag{A.6}$$

$$C_\theta(y|\Theta) = \frac{1}{n}\sum_{j=1}^{n} |WE_\theta(y|x_j,\Theta)| \tag{A.7}$$

We finally measure the contribution to a task t by selecting the contributions $C_\theta(y|\Theta)$ that satisfy $y = y_{true}$ which are the class predictions that are correct. Furthermore filtering out predictions that had been incorrectly inferred from the network, we can increase certainty that the inputs used to evaluate the contributions lead to high probability for the correct class and low everywhere else. We further generalize the contribution of θ to task t by averaging over the contributions to each class y_k within task t (see equation A.8).

$$TC_\theta(t|\Theta) = \frac{1}{K}\sum_{k=1}^{K} C_\theta(y_k|\Theta),\ t \in |Tasks|,\ K \in |outputs_t| \tag{A.8}$$