PocketNN is for integer-only training and inference of neural networks. It directly operates on integers without quantization.

Direct feedback alignment (DFA) was used instead of backpropagation (BP) for training; a family of new activation functions was designed for integer-only DNNs and named Pocket Activations.

PocketNN was implemented in pure C++ without any dependencies for maximum compatibility and portability.

It will be useful tool for tinyML researchers and developers working on integer-only DNNs.

Because $\delta_{DFA}^{[k]}$ is independent from $\delta_{DFA}^{[l]}$ of other layers, integer overflow is prevented in DFA.

PocketNN directly operates on integers without quantization.

Quantization is a conventional approach for integer-only DNNs. However, existing quantization algorithms have several drawbacks:

1. Many are for inference only; require floating-point operations on training.
2. Others support both training and inference. However, they involve complex operations such as bit shifting, scaling, and deterministic and stochastic roundings.
3. Customized fixed-point notations are often used to implement floating-point real numbers with integers. Conceptually, they are not using integers but still using floating-point numbers.
4. They often suffer from overflow during training.

PocketNN solves these problems by directly operating on integers without any explicit quantization.

Backpropagation (BP) causes integer overflow.

Backpropagation (BP), a de facto standard DNN training algorithm, can easily suffer from overflow in integer-only DNN training. Consider the $k^{th}$ layer of a fully connected neural network:

$\mathbf{h}^{[k]} = \mathbf{a}^{[k-1]}\mathbf{W}^{[k]} + \mathbf{b}^{[k]}$ and $\mathbf{a}^{[k]} = \text{actv}(\mathbf{h}^{[k]})$

BP updates the weight matrices and the bias vectors as below.

$\delta_{bp}^{[k]} = \dfrac{\partial J}{\partial \mathbf{W}^{[k]}} = \mathbf{a}^{[k-1]}\delta_{bp}^{[k-1]} \circ \text{actv}'(\mathbf{h}^{[k]})$ and $\dfrac{\partial J}{\partial \mathbf{b}^{[k]}} = \delta_{bp}^{[k]}$

The recursive multiplication of $\delta_{bp}s$ causes integer overflow.

Direct feedback alignment (DFA) can prevent overflow.

Direct feedback alignment (DFA) is a new emerging DNN training algorithm. DFA trains hidden layers independently from other layers by propagating error directly from the output layer to each individual hidden layer via fixed random feedback matrices. Again, consider the $k^{th}$ layer of a fully connected neural network:

$\mathbf{h}^{[k]} = \mathbf{a}^{[k-1]}\mathbf{W}^{[k]} + \mathbf{b}^{[k]}$ and $\mathbf{a}^{[k]} = \text{actv}(\mathbf{h}^{[k]})$

Instead of weights, appropriately sized random matrices $R^{[k]}$ are used to define $\delta_{DFA}^{[k]}$ such that

$\delta_{DFA}^{[k]} = \left(\mathbf{y} - \mathbf{y}^{[k]}\right) \circ \text{actv}'(\mathbf{h}^{[k]})$

Because $\delta_{DFA}^{[k]}$ is independent from other layers, integer overflow is prevented.