Abstract
Neuroorphic hardware is gaining momentum. In order to make neuroorphic hardware safe, it is necessary to understand fault tolerance of neural networks. In this project, we first show that this problem is hard, and then introduce the necessary assumptions to calculate this quantity. We test the bounds in proof-of-concept experiments. Our contribution is an algorithm for certifying achieving fault tolerance. Compared to other approaches, we have formal guarantees of fault tolerance.

Why fault tolerance?
Motivation: neurocomputing hardware can have crashes in neurons leading to a prediction mistake. The task of evaluating fault tolerance formally.

Sufficient conditions for fault tolerance
We overcome issues stated in the previous section by using additional assumptions. We will first bound the probability of too many neurochip crashes. Next, for the probable case, we will use a Taylor expansion.

Introduction
Fault tolerance was a popular line of research before the last 2 decades (1990s) because neural networks were expected to be implemented in neurocomputing hardware. However, researchers then limited to shallow architectures and their results are not applicable to modern networks. Now, some neurocomputing hardware is becoming popular again, which means that fault tolerance is an essential desirable property of neural networks. In this work, we study error propagation in neural networks and connect fault tolerance to other properties of a neural network such as robustness to adversarial examples and generalization properties.

Our model
Notations: (x, y) — training data point. x is the input vector. y is the label. Predictions are made as ŷ = max(Wx). Note that the last layer is linear.

Definition 1. Neural network: a function y = f(Wx) that maps inputs x ∈ R^n to outputs y ∈ R^m. The mapping could also fit that function’s derivatives which are bounded.

Definition 2. (Weight failure) NN: W_i is the weight failure if there exists an input x such that ŷ(x) = ŷ(x + δx) for all δx ∈ R^n.

Proposition 1. (NP-hardness) The task of computing fault tolerance is NP-hard in general. In addition, there are “pathological” cases for which even a small weight perturbation could lead to a large change in the output.

Hardness of the fault tolerance in the general case
We show first that the task of computing the fault tolerance is NP-hard in general. In addition, there are "pathological" cases for which even a small weight perturbation could lead to a large change in the output.

Conclusion
Fault crash tolerance is an overlooked concrete AI safety problem (1). We introduce a probabilistic framework to study fault tolerance. First, we show that the problem is hard in general. Next, we introduce the necessary assumptions to obtain results for a practical case of neurocomputing hardware: we design and test a new algorithm based on Proposition 5 to guarantee fault tolerance.