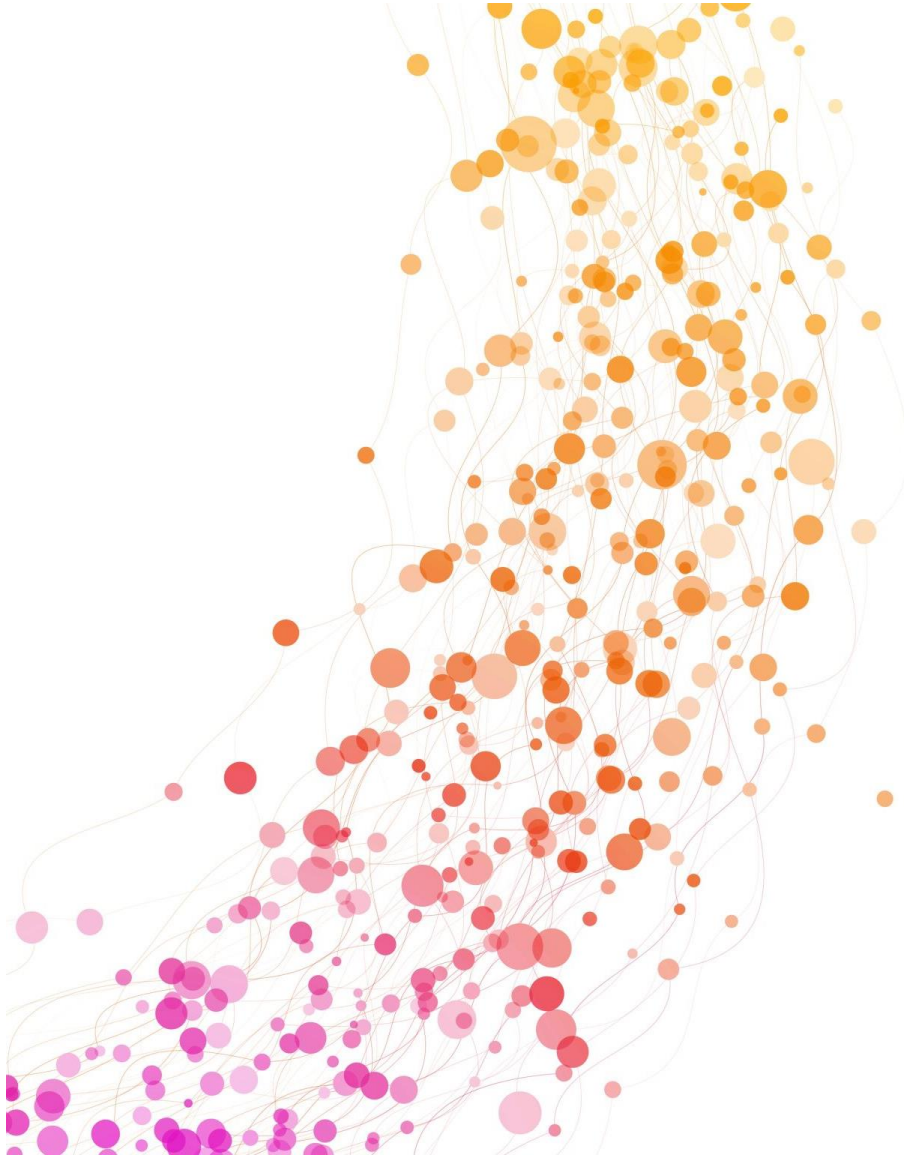
Conformance Testing of Machine Learning API



Chai Chaoweeraprasit

Development Lead Windows AI @ Microsoft

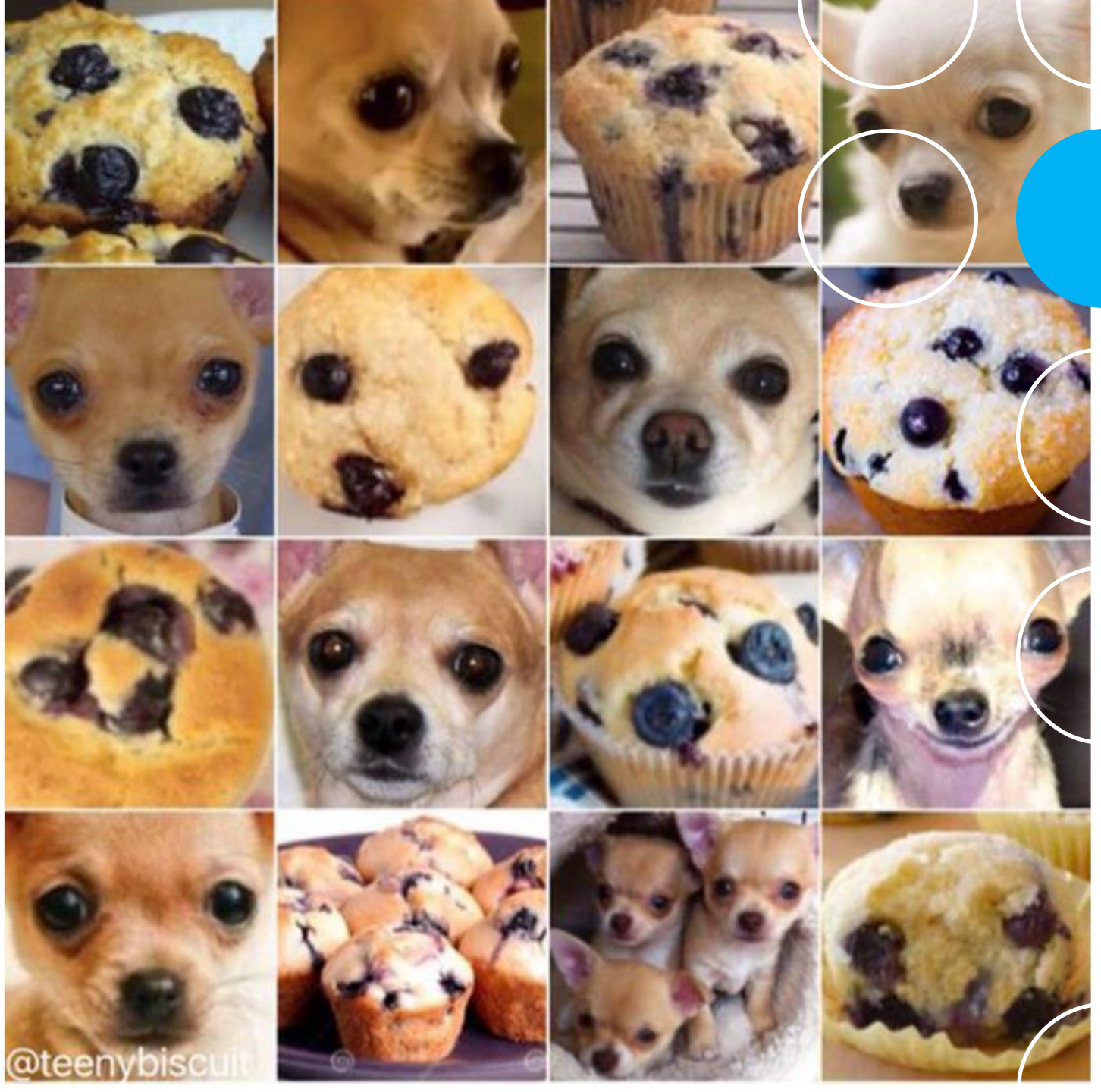
WebNN co-editor

Problems

- Modern ML models run on a wide variety of specialized hardware
- Most known AI use cases relies on floating-point calculations
- Processing of deep neural networks unavoidably accumulates floating-point errors

Is That a Muffin?

- Karen Zack posted a series of bizarre photo quiz on her Twitter account (March 2016)
- Her “Animals vs. Food” posts went viral and later became an AI challenge



Variability of Results

Precision differences	• Data types e.g. half, float, double
Hardware differences	• Architectural differences e.g. floating-point vs. fixed-point
Algorithmic differences	• Any two convolution algorithms are never alike
Numerical differences	• Non-deterministic computation or lossy conversions

Comparison Methods

Fuzzy comparison with epsilon introduces numerical differences on top of everything else

ULP *“unit of least precision”* is the distance between two consecutive floating-point values

ULP-based comparison removes numerical differences between the two values

ULP-based Compare

```
int64_t GetBitwise(float value) {
    int64_t bitwiseValue = (value < T(0)) ? ~int64_t(0) : 0; // Extend sign.
    *std::launder(reinterpret_cast<T*>(&bitwiseValue)) = value;
    return bitwiseValue;
}

bool CompareUlp(float a, float b, uint64_t ulp) {
    return static_cast<uint64_t>(abs(GetBitwise(a) - GetBitwise(b))) > ulp;
}
```

Baseline and Tolerances

- Algorithmic differences is unavoidable
- Tolerances are acceptable differences between the result *(what-is)* and the baseline value *(what-should-be)*
- ULP-based tolerances remove both the hardware and numerical differences during a comparison
- An *“ideal”* baseline is invariant and stable

Test Construction Strategy