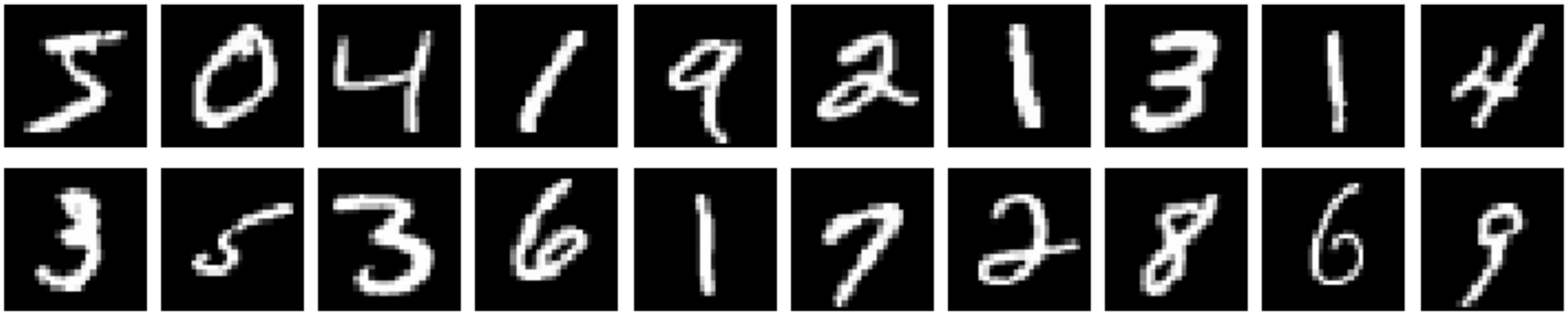
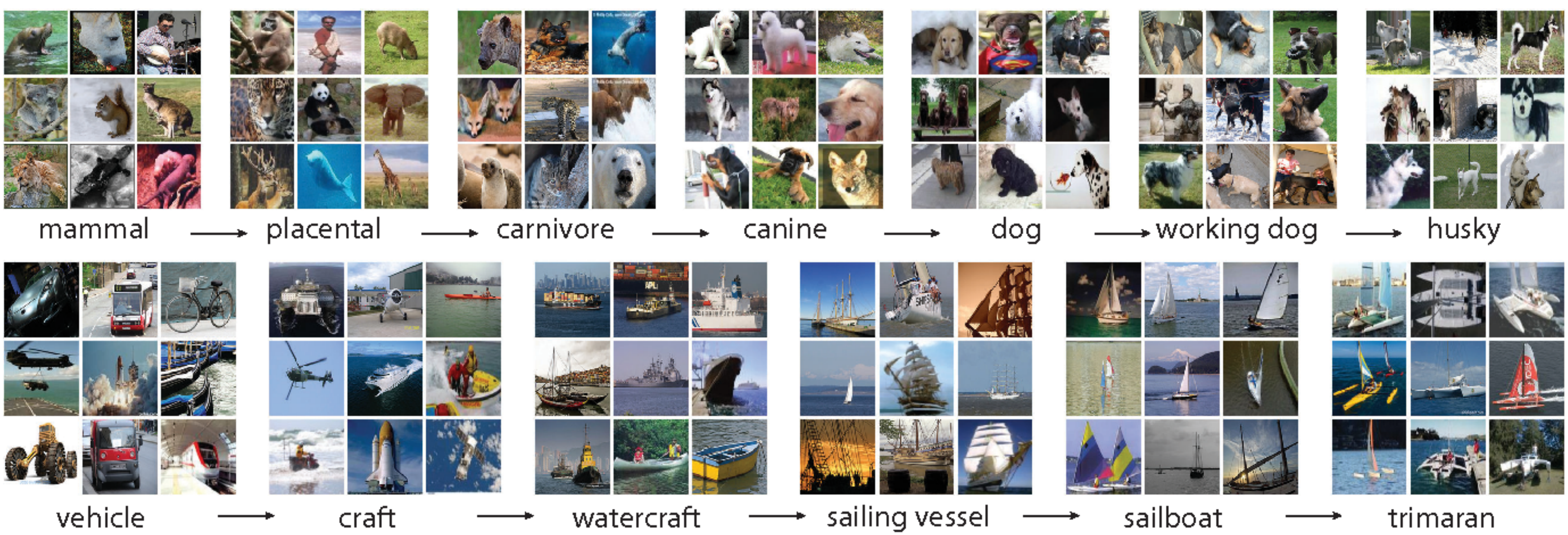
Adaptive Boosting (AdaBoost)

William Wei
National Center for Supercomputing Applications
Department of Physics
University of Illinois at Urbana-Champaign

General setting of the classification problem



Original source: http://yann.lecun.com/exdb/mnist/



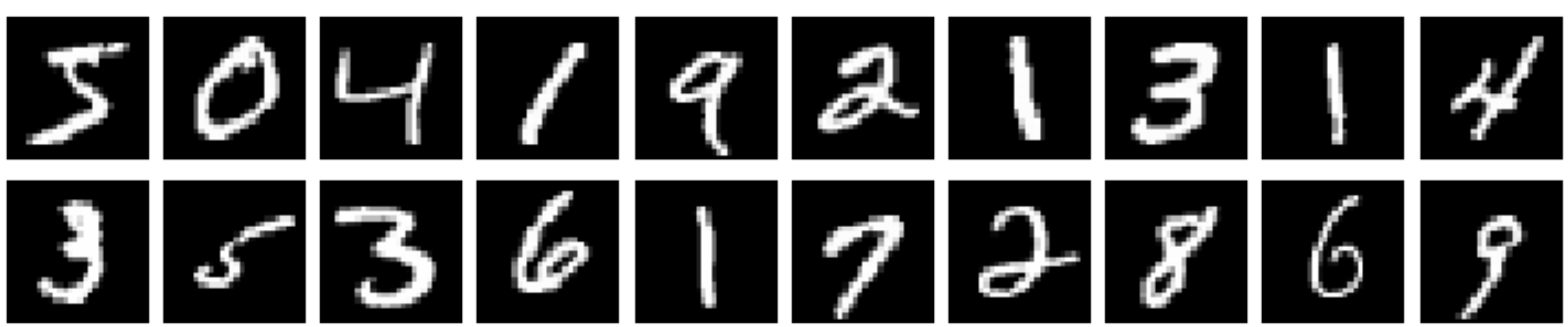
Original source: ImageNet
https://www.semanticscholar.org/paper/ImageNet%3A-A-large-scale-hierarchical-image-database-Deng-Dong/38211dc39e41273c0007889202c69f841e02248a

General setting of the classification problem

- **Observation** $\{X_i\}_{i=1}^N$
- **Hidden parameters (labels)** $\{Y_i\}_{i=1}^N$
- **Assume** (X, Y) **follows some fixed but unknown probability distribution** f_{XY}
- **Need to find a function** $g : X \to Y$ **(deterministic)**
  or $g : X \to \Delta_Y$ **(probabilistic)**

  such that $L(g) = \mathbf{E}[C(g(X), f_{Y|X}(\,\cdot\,|X))]$ **is minimizer given certain cost function** C

- **Optimal classifier:** $g^* = f_{Y|X}(Y|X)$
- **Since** f_{XY} **is unknown, need to approximate it.**



Original source: http://yann.lecun.com/exdb/mnist/

General setting of the classification problem

- **Consider a function class** g_W **parameterized by** $W \in \mathbb{R}^n$
- **Given a sequence of** $\{X_i, Y_i\}_{i=1}^N$

$$\hat{W} = \arg\min_W \frac{\sum_{i=1}^N C(g_W(X_i), Y_i, W)}{N}$$

- **If** $g_{\hat{W}} \approx f_{Y|X}(\cdot \mid X)$**, we say the model generalize well.**

Weighted linear combination of classifiers

For fixed base classifiers, $g_i(\cdot), i = 1,\ldots,N$, for binary classification, we can consider the following form,

$$g_w(\cdot) = sgn\left(\sum_{i=1}^{N} w_i g_i(\cdot)\right)$$

Need a strategy to combine base classifiers.

AdaBoost of Freund and Schapire (binary classification)

Let $\mathscr{G} = \{g_i\}_{i=1}^{N}$ **be a fixed set of weak classifiers, and,** $\{Z_i\}_{i=1}^{n}$

where, $Z_i = (X_i, Y_i)$ **, is the training set.** $X_i \in \mathbb{R}^n \quad Y_i \in \{-1, +1\}$

The AdaBoost algorithm works iteratively as follows:

- **Initialize** $w^{(1)} = \left(w_1^{(1)}, \ldots, w_n^{(1)}\right)$**, with** $w_i^{(1)} = \dfrac{1}{n}$
- **At each iteration k = 1,…,K:**

1. **Let** $g_k \in \mathscr{G}$ **be any weak learner that minimizes the weighted empirical error,**

$$e_k(g) := \sum_{i=1}^{n} w_i^{(k)} \mathbf{1}_{Y_i \neq g(X_i)}$$

The AdaBoost algorithm works iteratively as follows:

- **Initialize** $w^{(1)} = \left(w_1^{(1)}, \ldots, w_n^{(1)}\right)$**, with** $w_i^{(1)} = \frac{1}{n}$
- **At each iteration k = 1,…,K:**

1. Let $g_k \in \mathcal{G}$ **be any weak learner that minimizes the weighted empirical error,** $e_k(g) := \sum_{i=1}^{n} w_i^{(k)} \mathbf{1}_{Y_i \neq g(X_i)}$

2. $w_i^{(k+1)} = \dfrac{w_i^{(k)} \exp(-\alpha_k Y_i g_k(X_i))}{\mathcal{Z}_k}$ **, where** $\alpha_k = \frac{1}{2} \log \dfrac{1 - e_k}{e_k}$

And, $\mathcal{Z}_k = \sum_{i=1}^{n} w_i^{(k)} \exp(-\alpha_k Y_i g_k(X_i))$

- **After K iterations, output** $\hat{f}_n(x) := \dfrac{\sum_{i=1}^{K} \alpha_k g_k(x)}{\sum_{k=1}^{K} \alpha_k}$