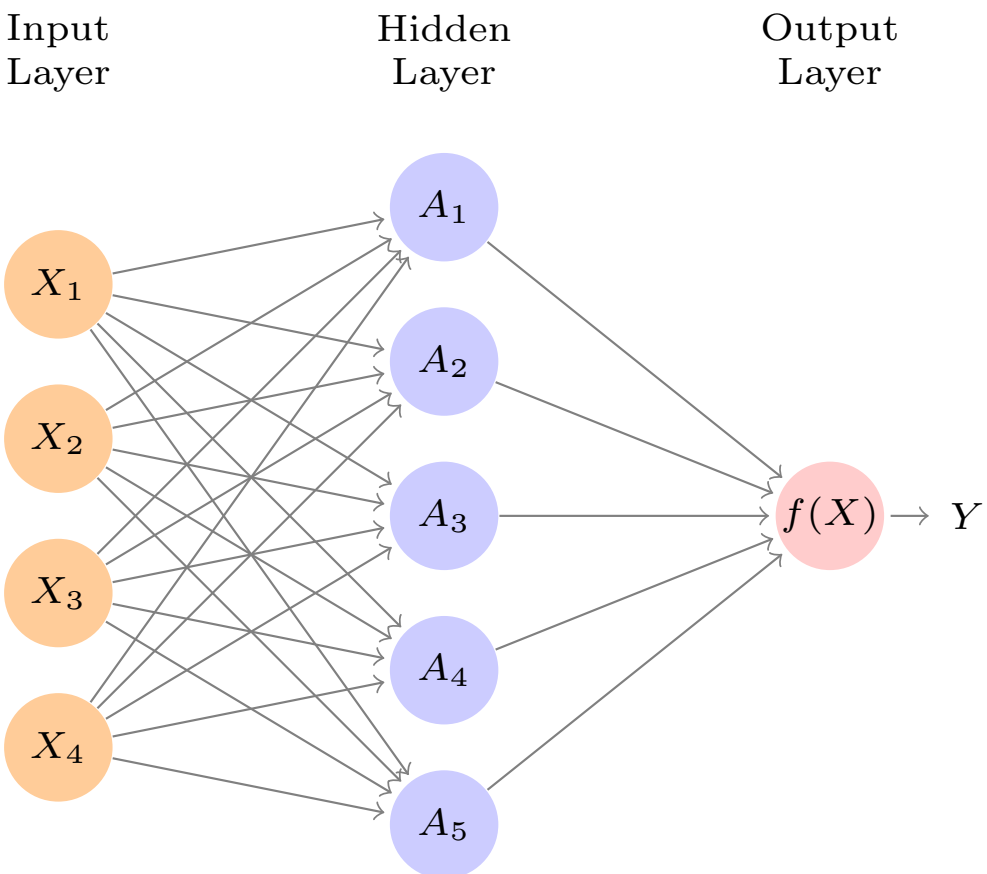
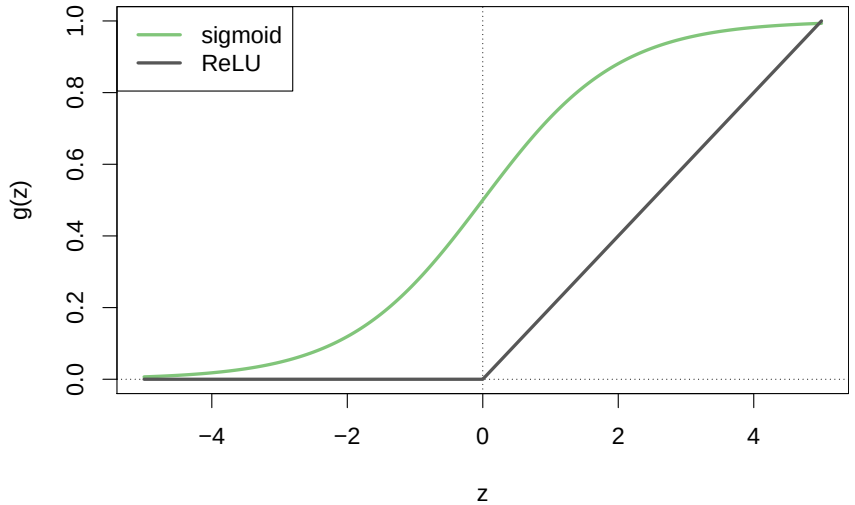
Single Layer Neural Network

$$
\begin{aligned}
f(X) &= \beta_0 + \sum_{k=1}^{K} \beta_k h_k(X) \\
&= \beta_0 + \sum_{k=1}^{K} \beta_k g(w_{k0} + \sum_{j=1}^{p} w_{kj} X_j).
\end{aligned}
$$



Details

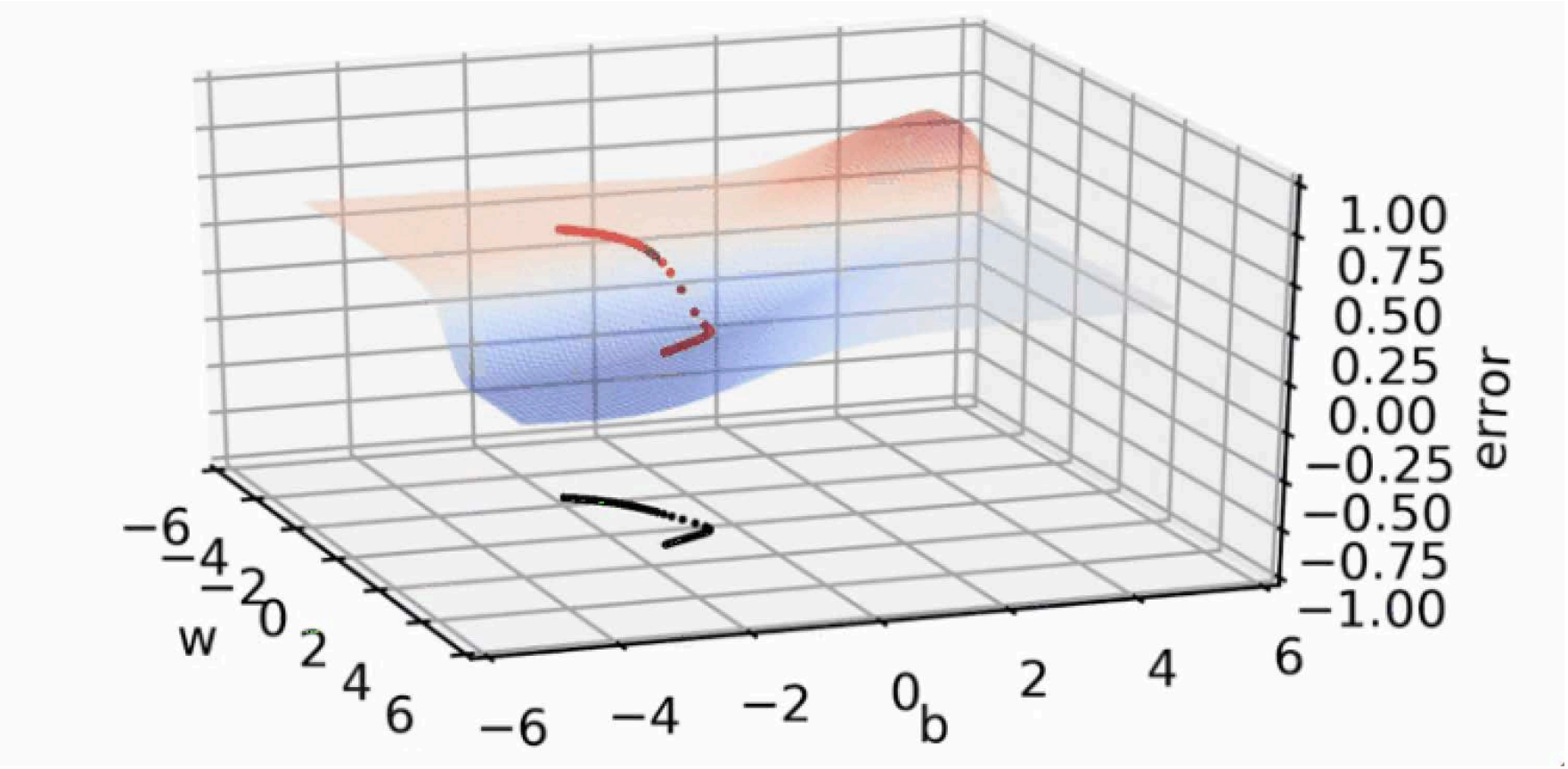


- $A_k = h_k(X) = g(w_{k0} + \sum_{j=1}^{p} w_{kj} X_j)$ are called the *activations* in the *hidden layer*.
- $g(z)$ is called the *activation function*. Popular are the *sigmoid* and *rectified linear*, shown in figure.

sigmoid: $$\sigma(x) = \frac{1}{1 + e^{-x}}$$

ReLU: $$f(x) = x^+ = \max(0, x) = \frac{x + |x|}{2} = \begin{cases} x & \text{if } x > 0, \\ 0 & \text{otherwise,} \end{cases}$$

Gradient Descent



Backpropagation

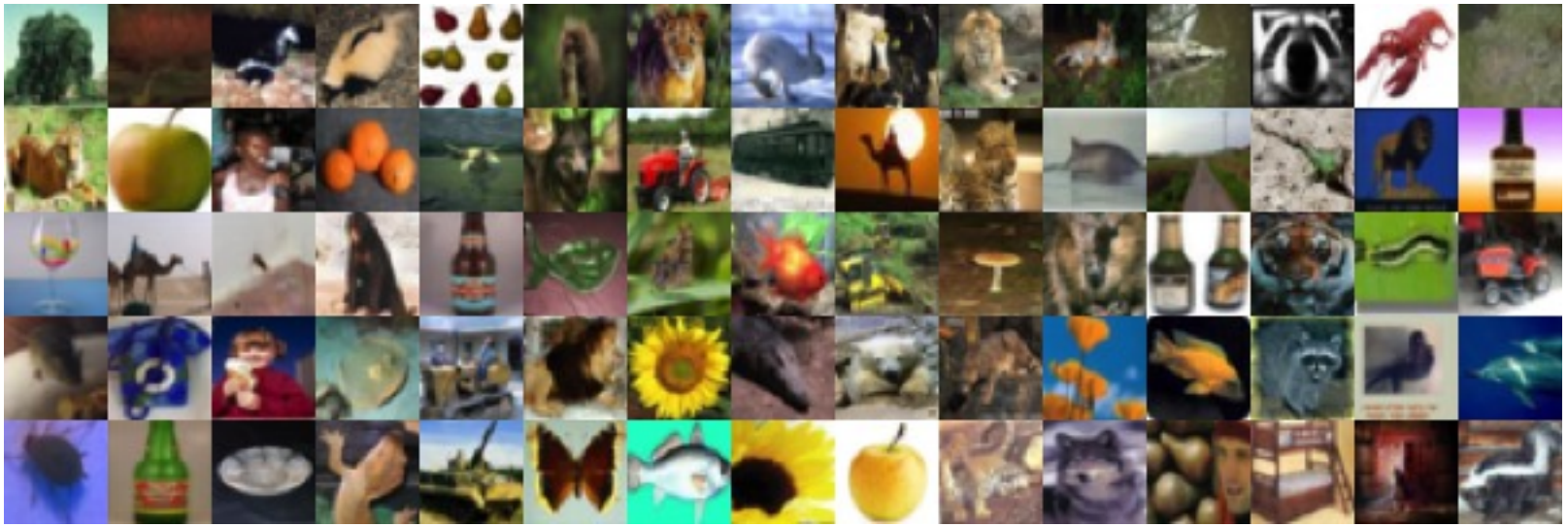
$$w_{t+1} = w_t - \eta \nabla w_t$$

$$b_{t+1} = b_t - \eta \nabla b_t$$

$$where \nabla w_t = \frac{\partial L(w,b)}{\partial w}_{w=w_t, b=b_t}, \nabla b_t = \frac{\partial L(w,b)}{\partial b}_{w=w_t, b=b_t}$$

Gradient Descent Update Rule

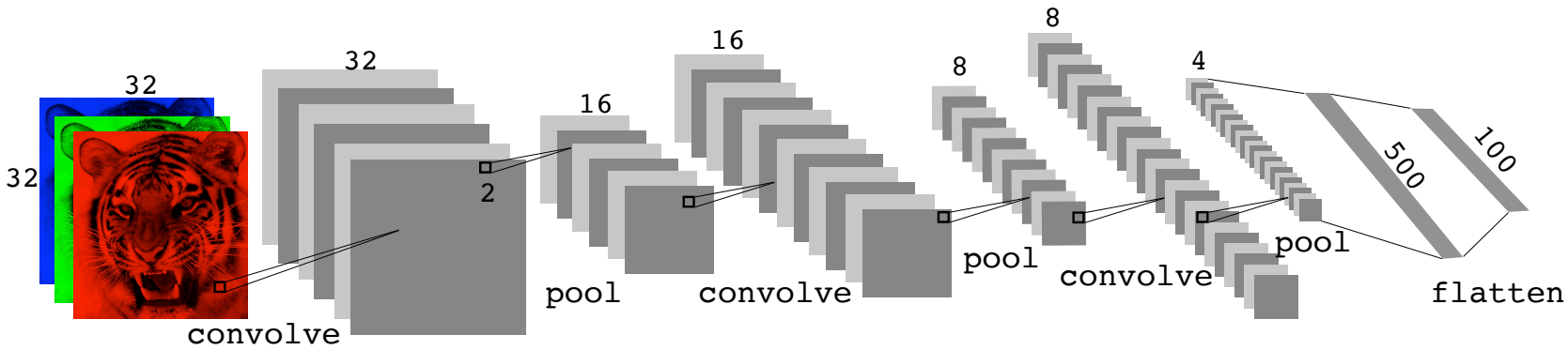
Convolutional Neural Network — CNN



- Major success story for classifying images.
- Shown are samples from `CIFAR100` database. 32×32 color natural images, with 100 classes.
- $50K$ training images, $10K$ test images.

Each image is a three-dimensional array or *feature map:* $32 \times 32 \times 3$ array of 8-bit numbers. The last dimension represents the three color channels for red, green and blue.

Architecture of a CNN



- Many convolve + pool layers.
- Filters are typically small, e.g. each channel 3×3.
- Each filter creates a new channel in convolution layer.
- As pooling reduces size, the number of filters/channels is typically increased.
- Number of layers can be very large. E.g. `resnet50` trained on `imagenet` 1000-class image data base has 50 layers!

代码演示

Deep Learning

In this section we demonstrate how to fit the examples discussed in the text. We use the `Python` {} `torch` package, along with the `pytorch_lightning` package which provides utilities to simplify fitting and evaluating models. This code can be impressively fast with certain special processors, such as Apple's new M1 chip. The package is well-structured, flexible, and will feel comfortable to `Python` {} users. A good companion is the site pytorch.org/tutorials. Much of our code is adapted from there, as well as the `pytorch_lightning` documentation. {The precise URLs at the time of writing are https://pytorch.org/tutorials/beginner/basics/intro.html and https://pytorch-lightning.readthedocs.io/en/latest/.}

We start with several standard imports that we have seen before.

•[1]:
```
import numpy     as np
import pandas    as pd
from matplotlib.pyplot    import subplots

from sklearn.linear_model import \
     (LinearRegression,
      LogisticRegression,
      Lasso)
from sklearn.preprocessing   import StandardScaler
from sklearn.model_selection import KFold
from sklearn.pipeline        import Pipeline
from sklearn.model_selection import \
     (train_test_split,
      GridSearchCV)

from ISLP                import load_data
from ISLP.models         import ModelSpec as MS
```

Torch-Specific Imports

There are a number of imports for `torch` . (These are not included with `ISLP` , so must be installed separately.) First we import the main library and essential tools used to specify sequentially-structured networks.

JAMES | Journal of Advances in Modeling Earth Systems®

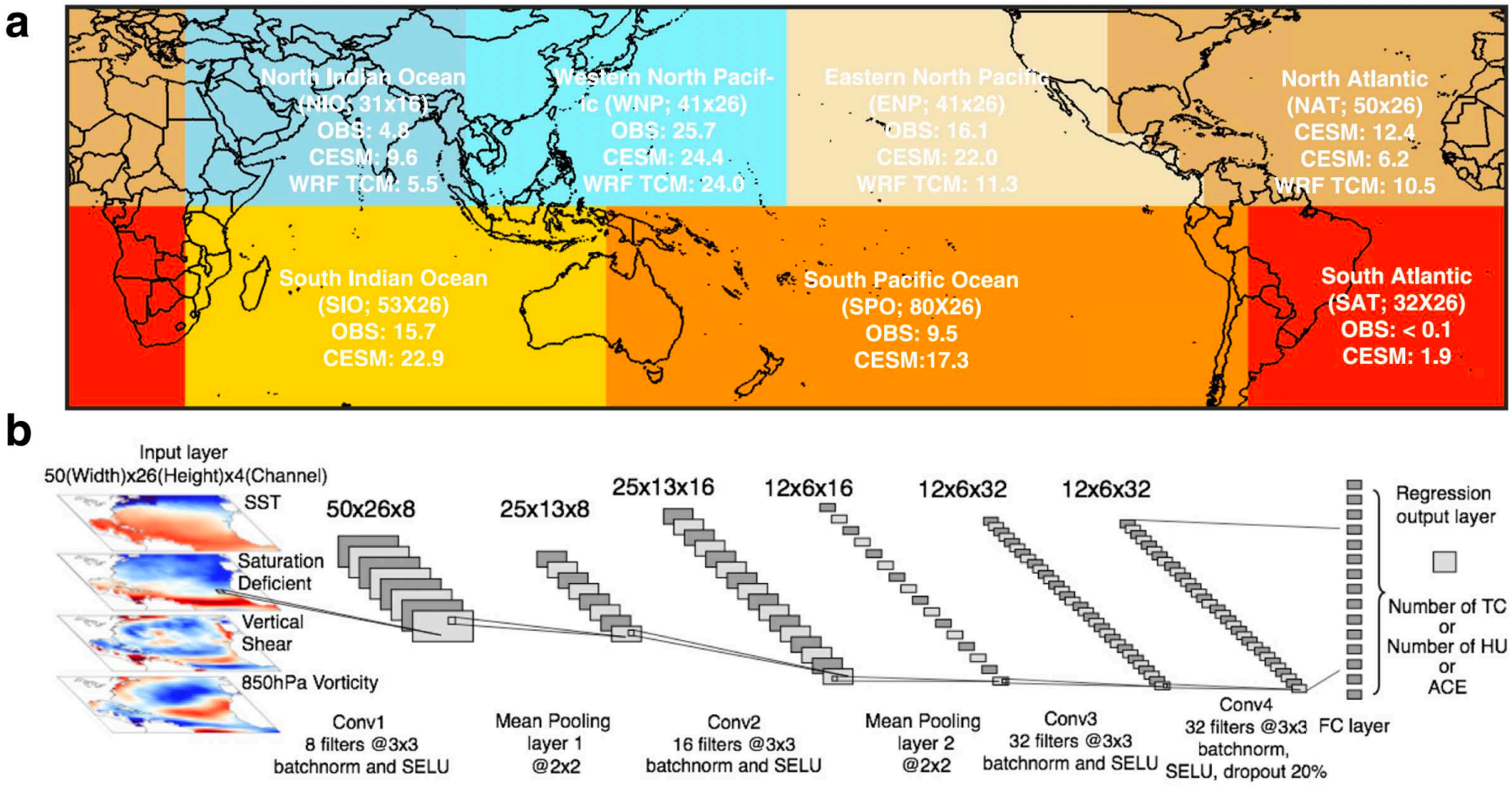
RESEARCH ARTICLE
10.1029/2022MS003596

Special Section:
Machine learning application to Earth system modeling

Using Convolutional Neural Network to Emulate Seasonal Tropical Cyclone Activity

Dan Fu[1], Ping Chang[1,2], and Xue Liu[1]

[1]Department of Oceanography, Texas A&M University, College Station, TX, USA, [2]Department of Atmospheric Sciences, Texas A&M University, College Station, TX, USA


a
North Indian Ocean
(NIO; 31x16)
OBS: 4.8
CESM: 9.6
WRF TCM: 5.5
Western North Pacif-
ic (WNP; 41x26)
OBS: 25.7
CESM: 24.4
WRF TCM: 24.0
Eastern North Pacific
(ENP; 41x26)
OBS: 16.1
CESM: 22.0
WRF TCM: 11.3
North Atlantic
(NAT; 50x26)
CESM: 12.4
CESM: 6.2
WRF TCM: 10.5
South Indian Ocean
(SIO; 53X26)
OBS: 15.7
CESM: 22.9
South Pacific Ocean
(SPO; 80X26)
OBS: 9.5
CESM:17.3
South Atlantic
(SAT; 32X26)
OBS: < 0.1
CESM: 1.9
b
Input layer
50(Width)x26(Height)x4(Channel)
SST
Saturation
Deficient
Vertical
Shear
850hPa Vorticity
50x26x8
25x13x8
25x13x16
12x6x16
12x6x32
12x6x32
Regression
output layer
Number of TC
or
Number of HU
or
ACE
Conv1
8 filters @3x3
batchnorm and SELU
Mean Pooling
layer 1
@2x2
Conv2
16 filters @3x3
batchnorm and SELU
Mean Pooling
layer 2
@2x2
Conv3
32 filters @3x3
batchnorm and SELU
Conv4
32 filters @3x3
batchnorm,
SELU, dropout 20%
FC layer