CS 4803 / 7643: Deep Learning

Topics:
- (Finish) Transfer Learning
- Recurrent Neural Networks

Zsolt Kira
Georgia Tech
Administrative

• HW3 will be out today
  – Due date pushed back to March 15th to allow you more time

• Next Thursday: FB lecture on attention/transformers
Part 1

- Activation Functions
- Data Preprocessing
- Weight Initialization
- Batch Normalization
- Babysitting the Learning Process
- Hyperparameter Optimization

Slide Credit: Fei-Fei Li, Justin Johnson, Serena Yeung, CS 231n
Step 1: Preprocess the data

In practice, you may also see **PCA** and **Whitening** of the data

Slide Credit: Fei-Fei Li, Justin Johnson, Serena Yeung, CS 231n
TLDR: In practice for Images: center only

- Subtract the mean image (e.g. AlexNet)
  (mean image = [32,32,3] array)
- Subtract per-channel mean (e.g. VGGNet)
  (mean along each channel = 3 numbers)

Not common to normalize variance, to do PCA or whitening
He et al., 2015
(note additional 2/)
Batch Normalization

- Improves gradient flow through the network
- Allows higher learning rates
- Reduces the strong dependence on initialization
- Acts as a form of regularization in a funny way, and slightly reduces the need for dropout, maybe

\[
\mu_B \leftarrow \frac{1}{m} \sum_{i=1}^{m} x_i \quad // \text{mini-batch mean}
\]

\[
\sigma_B^2 \leftarrow \frac{1}{m} \sum_{i=1}^{m} (x_i - \mu_B)^2 \quad // \text{mini-batch variance}
\]

\[
\hat{x}_i \leftarrow \frac{x_i - \mu_B}{\sqrt{\sigma_B^2 + \epsilon}} \quad // \text{normalize}
\]

\[
y_i \leftarrow \gamma \hat{x}_i + \beta \equiv \text{BN}_{\gamma, \beta}(x_i) \quad // \text{scale and shift}
\]

[Ioffe and Szegedy, 2015]

Slide Credit: Fei-Fei Li, Justin Johnson, Serena Yeung, CS 231n
Summary

We looked in detail at:

- Activation Functions (use ReLU)
- Data Preprocessing (images: subtract mean)
- Weight Initialization (use Xavier/He init)
- Batch Normalization (use)
- Babysitting the Learning process
- Hyperparameter Optimization (random sample hyperparams, in log space when appropriate)

TLDRs

Slide Credit: Fei-Fei Li, Justin Johnson, Serena Yeung, CS 231n
Model Ensembles: Tips and Tricks

Instead of training independent models, use multiple snapshots of a single model during training!

Huang et al, “Snapshot ensembles: train 1, get M for free”, ICLR 2017
Figures copyright Yixuan Li and Geoff Pleiss, 2017. Reproduced with permission.

Cyclic learning rate schedules can make this work even better!
Regularization: Dropout

\[ p = 0.5 \] # probability of keeping a unit active. higher = less dropout

def train_step(X):
    """ X contains the data """

    # forward pass for example 3-layer neural network
    H1 = np.maximum(0, np.dot(W1, X) + b1)
    U1 = np.random.rand(*H1.shape) < p # first dropout mask
    H1 *= U1 # drop!
    H2 = np.maximum(0, np.dot(W2, H1) + b2)
    U2 = np.random.rand(*H2.shape) < p # second dropout mask
    H2 *= U2 # drop!
    out = np.dot(W3, H2) + b3

    # backward pass: compute gradients... (not shown)
    # perform parameter update... (not shown)
Regularization: Data Augmentation

Load image and label → "cat" → Transform image → CNN → Compute loss

Slide Credit: Fei-Fei Li, Justin Johnson, Serena Yeung, CS 231n
Transfer Learning with CNNs

1. Train on Imagenet

2. Small Dataset (C classes)
   - Freeze these
   - Reinitialize this and train

3. Bigger dataset
   - Train these
   - With bigger dataset, train more layers
   - Fine-tune these
   - Lower learning rate when finetuning; 1/10 of original LR is good starting point

---

Slide Credit: Fei-Fei Li, Justin Johnson, Serena Yeung, CS 231n


<table>
<thead>
<tr>
<th>More generic</th>
<th>More specific</th>
</tr>
</thead>
<tbody>
<tr>
<td>MaxPool</td>
<td>Conv-64</td>
</tr>
<tr>
<td>Conv-512</td>
<td>Conv-64</td>
</tr>
<tr>
<td>Conv-512</td>
<td>Conv-128</td>
</tr>
<tr>
<td>MaxPool</td>
<td>Conv-128</td>
</tr>
<tr>
<td>Conv-256</td>
<td>Conv-128</td>
</tr>
<tr>
<td>Conv-256</td>
<td>MaxPool</td>
</tr>
<tr>
<td>Conv-512</td>
<td>Conv-512</td>
</tr>
<tr>
<td>Conv-512</td>
<td>MaxPool</td>
</tr>
<tr>
<td>Conv-64</td>
<td>Conv-64</td>
</tr>
<tr>
<td>Conv-64</td>
<td>Conv-64</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>very similar dataset</th>
<th>very different dataset</th>
</tr>
</thead>
<tbody>
<tr>
<td>very little data</td>
<td>?</td>
</tr>
<tr>
<td>quite a lot of data</td>
<td>?</td>
</tr>
<tr>
<td>More specific</td>
<td>More generic</td>
</tr>
<tr>
<td>---------------</td>
<td>--------------</td>
</tr>
<tr>
<td><strong>very little data</strong></td>
<td>Use Linear Classifier on top layer</td>
</tr>
<tr>
<td><strong>quite a lot of data</strong></td>
<td>Finetune a few layers</td>
</tr>
</tbody>
</table>

very similar dataset

very different dataset

---

Slide Credit: Fei-Fei Li, Justin Johnson, Serena Yeung, CS 231n
<table>
<thead>
<tr>
<th>More specific</th>
<th>More generic</th>
<th>very similar dataset</th>
<th>very different dataset</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>very little data</strong></td>
<td></td>
<td>Use Linear Classifier on top layer</td>
<td>You’re in trouble… Try linear classifier from different stages</td>
</tr>
<tr>
<td><strong>quite a lot of data</strong></td>
<td></td>
<td>Finetune a few layers</td>
<td>Finetune a larger number of layers</td>
</tr>
</tbody>
</table>

---

Slide Credit: Fei-Fei Li, Justin Johnson, Serena Yeung, CS 231n
Transfer learning with CNNs is pervasive…
(it’s the norm, not an exception)

Object Detection
(Fast R-CNN)

Figure copyright Ross Girshick, 2015. Reproduced with permission.

Image Captioning: CNN + RNN

Figure copyright IEEE, 2015. Reproduced for educational purposes.
Transfer learning with CNNs is pervasive… (it’s the norm, not an exception)

Object Detection (Fast R-CNN)

CNN pretrained on ImageNet

Image Captioning: CNN + RNN

Figure copyright Ross Girshick, 2015. Reproduced with permission.

Figure copyright IEEE, 2015. Reproduced for educational purposes.
Transfer learning with CNNs is pervasive…
(it’s the norm, not an exception)

Object Detection
(Fast R-CNN)

CNN pretrained on ImageNet

Image Captioning: CNN + RNN

Word vectors pretrained with word2vec

Figure copyright Ross Girshick, 2015. Reproduced with permission.

Figure copyright IEEE, 2015. Reproduced for educational purposes.
The most effective method: Gather more data!

Revisiting the Unreasonable Effectiveness of Data

Deep Learning Scaling is Predictable, Empirically
Joel Hestness, Sharan Narang, Newsha Ardalani, Gregory Diamos, Heewoo Jun, Hassan Kianinejad, Md. Mostofa Ali Patwary, Yang Yang, Yanqi Zhou
Takeaway for your projects and beyond:
Have some dataset of interest but it has < ~1M images?

1. Find a very large dataset that has similar data, train a big ConvNet there
2. Transfer learn to your dataset

Deep learning frameworks provide a “Model Zoo” of pretrained models so you don’t need to train your own

Caffe: https://github.com/BVLC/caffe/wiki/Model-Zoo
TensorFlow: https://github.com/tensorflow/models
PyTorch: https://github.com/pytorch/vision
Life is never so simple

There are several areas being researched
- Batch size
- Regularization and generalization
- Overparameterization and why SGD is so good

Why is this still not understood?
- Our understanding comes from built-in intuition that is repeated but not always tested
- Difficult to apply theory being developed
Plan for Today

• Recurrent Neural Networks (RNNs)
  – Example Problem: (Character-level) Language modeling
  – Learning: (Truncated) BackProp Through Time (BPTT)
  – Visualizing RNNs
  – Example: Image Captioning
  – Inference: Beam Search
  – Multilayer RNNs
  – Problems with gradients in “vanilla” RNNs
  – LSTMs (and other RNN variants)
New Topic: RNNs
Why model sequences?

Figure Credit: Carlos Guestrin
Why model sequences?
Sequences are everywhere…

Foreign minister.

FOREIGN MINISTER.

THE SOUND OF

\[ x = \begin{array}{ccccccc}
  a_1 & = & 2 & \quad a_2 & = & 0 & \quad a_3 & = & 1 \\
  a_4 & = & 3 & \quad a_5 & = & 4 & \quad a_6 & = & 2 \\
  a_7 & = & 5
\end{array} \]

\[ y = \begin{array}{cccc}
  & \text{please} & \text{return} & \text{the} & \text{car} \\
\end{array} \]

(C) Dhruv Batra

Image Credit: Alex Graves and Kevin Gimpel
Even where you might not expect a sequence…

Classify images by taking a series of “glimpses”
Even where you might not expect a sequence...

- Output ordering = sequence

(Image Credit: Ba et al.; Gregor et al)

(C) Dhruv Batra
Sequences in Input or Output?

- It’s a spectrum…

```plaintext
Input: No sequence  
Output: No sequence  
Example: “standard” classification / regression problems
```

(C) Dhruv Batra
Sequences in Input or Output?

• It’s a spectrum…

Input: No sequence
Output: No sequence
Example: "standard" classification / regression problems

Input: No sequence
Output: Sequence
Example: Im2Caption
Sequences in Input or Output?

• It’s a spectrum…

- **One to one**
  - Input: No sequence
  - Output: No sequence
  - Example: “standard” classification / regression problems

- **One to many**
  - Input: No sequence
  - Output: Sequence
  - Example: Im2Caption

- **Many to one**
  - Input: Sequence
  - Output: No sequence
  - Example: sentence classification, multiple-choice question answering

Image Credit: Andrej Karpathy
Sequences in Input or Output?

- It’s a spectrum…

Input: No sequence
Output: No sequence
Example: “standard” classification / regression problems

Input: No sequence
Output: Sequence
Example: Im2Caption

Input: Sequence
Output: No sequence
Example: sentence classification, multiple-choice question answering

Input: Sequence
Output: Sequence
Example: machine translation, video classification, video captioning, open-ended question answering
(Non-Deep) Ways to deal with sequence labelling

- **Autoregressive models**
  - Predict the next term in a sequence from a fixed number of previous terms using delay taps.
  - 1st-order Autoregressive model, AR(1): \( y_t = w_0 + w_1y_{t-1} + \epsilon_t \)
  - 2nd-order Autoregressive model, AR(2): \( y_t = w_0 + w_1y_{t-1} + w_2y_{t-2} + \epsilon_t \)
  - And so on.

- **Hidden Markov Model, HMM**
  - HMMs have a discrete one-of-N hidden state. Transitions between states are stochastic and controlled by a transition probability matrix. Also, the outputs produced by a state are also stochastic, and are controlled by emission probabilities.
    - We can not be sure which state produced a given output. So, the state is “hidden”
    - It is easy to represent a probability distribution across the N states with N probabilities.
  - To predict the next output we need to infer the probability distribution over hidden states. HMMs have efficient algorithm for inference and learning.
What’s wrong with MLPs?

• Problem 1: Can’t model sequences
  – Fixed-sized Inputs & Outputs
  – No temporal structure
What’s wrong with MLPs?

• Problem 1: Can’t model sequences
  – Fixed-sized Inputs & Outputs
  – No temporal structure

• Problem 2: Pure feed-forward processing
  – No “memory”, no feedback
A Naïve Way to Do This
2 Key Ideas

• Parameter Sharing
  – in computation graphs = adding gradients
Computational Graph
Gradients add at branches
2 Key Ideas

• Parameter Sharing
  – in computation graphs = adding gradients

• “Unrolling”
  – in computation graphs with parameter sharing
How do we model sequences?

• No input

\[ s_t = f_\theta(s_{t-1}) \]
How do we model sequences?

• With inputs

\[ s_t = f_\theta(s_{t-1}, x_t) \]
2 Key Ideas

• Parameter Sharing
  – in computation graphs = adding gradients

• “Unrolling”
  – in computation graphs with parameter sharing

• Parameter sharing + Unrolling
  – Allows modeling arbitrary sequence lengths!
  – Keeps numbers of parameters in check
New Words

• Recurrent Neural Networks (RNNs)

• Recursive Neural Networks
  – General family; think graphs instead of chains

• Types:
  – “Vanilla” RNNs (Elman Networks)
  – Long Short Term Memory (LSTMs)
  – Gated Recurrent Units (GRUs)
  – …

• Algorithms
  – BackProp Through Time (BPTT)
  – BackProp Through Structure (BPTS)
Recurrent Neural Network
Recurrent Neural Network

usually want to predict a vector at some time steps
We can process a sequence of vectors $x$ by applying a recurrence formula at every time step:

$$h_t = f_W(h_{t-1}, x_t)$$

new state \hspace{1cm} old state \hspace{1cm} input vector at some time step

some function with parameters $W$
Recurrent Neural Network

We can process a sequence of vectors $\mathbf{x}$ by applying a recurrence formula at every time step:

$$h_t = f_W(h_{t-1}, x_t)$$

Notice: the same function and the same set of parameters are used at every time step.
(Vanilla) Recurrent Neural Network

The state consists of a single “hidden” vector $h$:

$$y_t = W_{hy} h_t + b_y$$

$$h_t = f_W(h_{t-1}, x_t)$$

$$h_t = \tanh(W_{hh} h_{t-1} + W_{xh} x_t + b_h)$$

Sometimes called a “Vanilla RNN” or an “Elman RNN” after Prof. Jeffrey Elman

Slide Credit: Fei-Fei Li, Justin Johnson, Serena Yeung, CS 231n
RNN: Computational Graph

Slide Credit: Fei-Fei Li, Justin Johnson, Serena Yeung, CS 231n
RNN: Computational Graph

Slide Credit: Fei-Fei Li, Justin Johnson, Serena Yeung, CS 231n
RNN: Computational Graph
RNN: Computational Graph

Re-use the same weight matrix at every time-step

Slide Credit: Fei-Fei Li, Justin Johnson, Serena Yeung, CS 231n
RNN: Computational Graph: Many to Many

Slide Credit: Fei-Fei Li, Justin Johnson, Serena Yeung, CS 231n
RNN: Computational Graph: Many to Many

$h_0 \rightarrow f_W \rightarrow h_1 \rightarrow f_W \rightarrow h_2 \rightarrow f_W \rightarrow h_3 \rightarrow \ldots \rightarrow h_T$

$y_1 \rightarrow L_1 \rightarrow y_2 \rightarrow L_2 \rightarrow y_3 \rightarrow L_3 \rightarrow y_T \rightarrow L_T$

Slide Credit: Fei-Fei Li, Justin Johnson, Serena Yeung, CS 231n
RNN: Computational Graph: Many to Many

Slide Credit: Fei-Fei Li, Justin Johnson, Serena Yeung, CS 231n
RNN: Computational Graph: Many to One

Slide Credit: Fei-Fei Li, Justin Johnson, Serena Yeung, CS 231n
RNN: Computational Graph: One to Many

Slide Credit: Fei-Fei Li, Justin Johnson, Serena Yeung, CS 231n
Sequence to Sequence: Many-to-one + one-to-many

**Many to one**: Encode input sequence in a single vector

Slide Credit: Fei-Fei Li, Justin Johnson, Serena Yeung, CS 231n
Sequence to Sequence: Many-to-one + one-to-many

**Many to one**: Encode input sequence in a single vector

**One to many**: Produce output sequence from single input vector

Slide Credit: Fei-Fei Li, Justin Johnson, Serena Yeung, CS 231n
Example:
Character-level Language Model

Vocabulary:
[h,e,l,o]

Example training sequence:
“hello”
Example:
Character-level Language Model

Vocabulary:
[h,e,l,o]

Example training sequence:
“hello”

\[
h_t = \tanh(W_{hh}h_{t-1} + W_{xh}x_t + b_h)\]
Distributed Representations Toy Example

- Local vs Distributed

(a)

no pattern

\[
\begin{array}{cccccc}
\_ & \_ & \_ & \_ & \_ & \_ \\
\end{array}
\]


\[
\begin{array}{cccccc}
\_ & \_ & \_ & \_ & \_ & \_ \\
\cdot & \_ & \_ & \_ & \_ & \_ \\
\_ & \_ & \_ & \_ & \_ & \_ \\
\_ & \_ & \_ & \_ & \_ & \_ \\
\_ & \_ & \_ & \_ & \_ & \_ \\
\end{array}
\]

\[
\begin{array}{cccccc}
\_ & \_ & \_ & \_ & \_ & \_ \\
\_ & \_ & \_ & \_ & \_ & \_ \\
\_ & \_ & \_ & \_ & \_ & \_ \\
\_ & \_ & \_ & \_ & \_ & \_ \\
\cdot & \_ & \_ & \_ & \_ & \_ \\
\end{array}
\]

\[
\begin{array}{cccccc}
\_ & \_ & \_ & \_ & \_ & \_ \\
\_ & \_ & \_ & \_ & \_ & \_ \\
\_ & \_ & \_ & \_ & \_ & \_ \\
\cdot & \_ & \_ & \_ & \_ & \_ \\
\cdot & \_ & \_ & \_ & \_ & \_ \\
\end{array}
\]
Distributed Representations Toy Example

- Can we interpret each dimension?
Power of distributed representations!

Local: $\bullet \quad \bullet \quad \bigcirc \quad \bullet = VR + HR + HE = ?$

Distributed: $\bullet \quad \bullet \quad \bigcirc \quad \bullet = V + H + E \approx \bigcirc$
Example:
Character-level Language Model

Vocabulary: [h,e,i,o]

Example training sequence: “hello”
Training Time: MLE / “Teacher Forcing”

Example:
Character-level Language Model

Vocabulary: [h,e,l,o]

Example training sequence: “hello”
Test Time: Sample / Argmax / Beam Search

Example:
Character-level Language Model Sampling

Vocabulary:
[h,e,l,o]

At test-time sample characters one at a time, feed back to model

Slide Credit: Fei-Fei Li, Justin Johnson, Serena Yeung, CS 231n
Test Time: Sample / Argmax / Beam Search

Example:
Character-level Language Model Sampling

Vocabulary:
[h,e,l,o]

At test-time sample characters one at a time, feed back to model

Slide Credit: Fei-Fei Li, Justin Johnson, Serena Yeung, CS 231n
Test Time: Sample / Argmax / Beam Search

Example:
Character-level Language Model Sampling

Vocabulary:
[h,e,l,o]

At test-time sample characters one at a time, feed back to model

Slide Credit: Fei-Fei Li, Justin Johnson, Serena Yeung, CS 231n
Example:
Character-level Language Model Sampling

Vocabulary: [h,e,l,o]

At test-time sample characters one at a time, feed back to model.
Let's do Monday.

Monday works for me.

Either day works for me.

Reply

Reply all

Forward
Backpropagation through time

Forward through entire sequence to compute loss, then backward through entire sequence to compute gradient
Truncated Backpropagation through time

Run forward and backward through chunks of the sequence instead of whole sequence
Truncated Backpropagation through time

Carry hidden states forward in time forever, but only backpropagate for some smaller number of steps

Slide Credit: Fei-Fei Li, Justin Johnson, Serena Yeung, CS 231n
Truncated Backpropagation through time
THE SONNETS

by William Shakespeare

From fairest creatures we desire increase,
That thereby beauty's rose might never die,
But as the riper should by time decease,
His tender heir might bear his memory:
But thou, contracted to thine own bright eyes,
Feed'st thy light's flame with self-substantial fuel,
Making a famine where abundance lies,
Thyself thy foe, to thy sweet self too cruel:
Thou art now the world's fresh ornament,
And only herald to the gaudy spring,
Within thine own bud buried thy content,
And tender churl mak'st waster in niggarding:
Pry the world, or else this glutton be,
To eat the world's due, by the grave and thee.

When forty winters shall besiege thy brow,
And dig deep trenches in thy beauty's field,
Thy youth's proud livery so gazed on now,
Will be aantee'd weed of small worth held:
Then being asked, where all thy beauty lies,
Where all the treasure of thy lusty days;
To say, within thine own deep sunken eyes,
Were an all-eating shame, and thriftless praise.
How much more praise deserv'd thy beauty's use,
If thou couldst answer 'This fair child of mine
Shall sum my count, and make my old excuse,'
Proving his beauty by succession thine!
This were to be new made when thou art old,
And see thy blood warm when thou feel'st it cold.
at first:

"Tmont thithey" fomesscerliund
Keushey. Thom here
sheulke, anmerenith ol sivh I lalterthend Bleipile shuwy fil on aseterlome
canoiognc Phe lism thond hon at. MeiDimorotion in ther thize."

"Why do what that day," replied Natasha, and wishing to himself the fact the
princess, Princess Mary was easier, fed in had oftened him.
Pierre aking his soul came to the packs and drove up his father-in-law women.
PANDARUS:
Alas, I think he shall be come approached and the day
When little strain would be attain'd into being never fed,
And who is but a chain and subjects of his death,
I should not sleep.

Second Senator:
They are away this miseries, produced upon my soul,
Breaking and strongly should be buried, when I perish
The earth and thoughts of many states.

DUKE VINCENTIO:
Well, your wit is in the care of side and that.

Second Lord:
They would be ruled after this chamber, and
my fair nues begun out of the fact, to be conveyed,
Whose noble souls I'll have the heart of the wars.

Clown:
Come, sir, I will make did behold your worship.

VIOLA:
I'll drink it.

VIOLA:
Why, Salisbury must find his flesh and thought
That which I am not aps, not a man and in fire,
To show the reining of the raven and the wars
To grace my hand reproach within, and not a fair are hand,
That Caesar and my goodly father's world;
When I was heaven of presence and our fleets,
We spare with hours, but cut thy council I am great,
Murdered and by thy master's ready there
My power to give thee but so much as hell:
Some service in the noble bondman here,
Would show him to her wine.

KING LEAR:
O, if you were a feeble sight, the courtesy of your law,
Your sight and several breath, will wear the gods
With his heads, and my hands are wonder'd at the deeds,
So drop upon your lordship's head, and your opinion
Shall be against your honour.
The Stacks Project: open source algebraic geometry textbook

Slide Credit: Fei-Fei Li, Justin Johnson, Serena Yeung, CS 231n
For $\bigoplus_{n=1,...,m}$ where $L_m = 0$, hence we can find a closed subset $H$ in $H$ and any sets $F$ on $X, U$ is a closed immersion of $S$, then $U \to T$ is a separated algebraic space.

**Proof.** Proof of (1). It also start we get

$$S = \text{Spec}(R) = U \times_X U \times_X U$$

and the comparichy in the fibre product covering we have to prove the lemma generated by $\prod Z \times_U U \to V$. Consider the maps $M$ along the set of points $\text{Sch}_{fppf}$ and $U \to U$ is the fibre category of $S$ in $U$ in Section ?? and the fact that any $U$ affine, see Morphisms, Lemma ???. Hence we obtain a scheme $S$ and any open subset $W \subset U$ in $\text{Sh}(G)$ such that $\text{Spec}(R') \to S$ is smooth or an

$$U = \bigcup U_i \times_S U_i$$

which has a nonzero morphism we may assume that $f_i$ is of finite presentation over $S$. We claim that $O_{X,x}$ is a scheme where $x, x', s' \in S'$ such that $O_{X,x} \to O_{X',x'}$, is separated. By Algebra, Lemma ?? we can define a map of complexes $GLS(x'/x'')$ and we win. □

To prove study we see that $F_{U'}$ is a covering of $X'$, and $T_i$ is an object of $F_{X/S}$ for $i > 0$ and $F_p$ exists and let $T_i$ be a presheaf of $O_X$-modules on $C$ as a $F$-module. In particular $\mathcal{F} = U/F$ we have to show that

$$M^* = T^* \otimes_{\text{Spec}(k)} O_{S,s} - i_X^{-1}F$$

is a unique morphism of algebraic stacks. Note that

$$\text{Arrows} = \text{(Sch/S)}_{fppf}^{\text{op}} \otimes_{\text{Sch}/S_{fppf}} (\text{Sch/S})_{fppf}$$

and

$$V = \Gamma(S, O) \hookrightarrow (U, \text{Spec}(A))$$

is an open subset of $X$. Thus $U$ is affine. This is a continuous map of $X$ is the inverse, the groupoid scheme $S$.

**Proof.** See discussion of sheaves of sets. □

The result for prove any open covering follows from the less of Example ???. It may replace $S$ by $X_{\text{spaces, etale}}$ which gives an open subspace of $X$ and $T$ equal to $S_{\text{zar}}$, see Descent, Lemma ???. Namely, by Lemma ?? we see that $R$ is geometrically regular over $S$.

---

**Lemma 0.1.** Assume (3) and (3) by the construction in the description. Suppose $X = \lim |X|$ (by the formal open covering $X$ and a single map $\text{Proj}_X(A) = \text{Spec}(B)$ over $U$ compatible with the complex

$$\text{Set}(A) = \text{Hom}(X, O_{X, \overline{X}})$$

When in this case of to show that $Q \to C_{/X}$ is stable under the following result in the second conditions of (1), and (3). This finishes the proof. By Definition ?? (without element is when the closed subschemes are catenary. If $T$ is surjective we may assume that $T$ is connected with residue fields of $S$. Moreover there exists a closed subspace $Z \subset X$ of $X$ where $U$ in $X'$ is proper (some defining as a closed subset of the uniqueness it suffices to check the fact that the following theorem

(1) $f$ is locally of finite type. Since $S = \text{Spec}(R)$ and $Y = \text{Spec}(R)$.

**Proof.** This is form all sheaves of sheaves on $X$. But given a scheme $U$ and a surjective étale morphism $U \to X$. Let $U \cap U = \prod_{i=1,...,n} U_i$ be the scheme $X / S$ at the schemes $X_i \to X$ and $U = \lim_{i} X_i$. □

The following lemma surjective restecomposes of this implies that $F_{X_0} = F_{X_0} = F_{X_0}$.

**Lemma 0.2.** Let $X$ be a locally Noetherian scheme over $S$, $E = F_{X/S}$. Set $T = J_1 \subset T_1$. Since $T_1 \subset T'$ are nonzero over $i_0 \leq p$ is a subset of $J_{n,0} \otimes \bar{A}_2$ works.

**Lemma 0.3.** In Situation ???. Hence we may assume $q' = 0$.

**Proof.** We will use the property we see that $p$ is the next functor (??). On the other hand, by Lemma ?? we see that

$$D(\text{O}_{X'}) = \text{O}_X(D)$$

where $K$ is an $F$-algebra where $\delta_{n+1}$ is a scheme over $S$. □
Proof. Omitted.

**Lemma 0.1.** Let $\mathcal{C}$ be a set of the construction.

Let $\mathcal{C}$ be a gerber covering. Let $\mathcal{F}$ be a quasi-coherent sheaves of $\mathcal{O}$-modules. We have to show that

$$\mathcal{O}_{\mathcal{O}_X} = \mathcal{O}_X(\mathcal{C})$$

Proof. This is an algebraic space with the composition of sheaves $\mathcal{F}$ on $\mathcal{X}_{\text{etale}}$ we have

$$\mathcal{O}_X(\mathcal{F}) = \{\text{morph}_1 \times_{\mathcal{O}_X} (\mathcal{G}, \mathcal{F})\}$$

where $\mathcal{G}$ defines an isomorphism $\mathcal{F} \to \mathcal{F}$ of $\mathcal{O}$-modules.

**Lemma 0.2.** This is an integer $\mathcal{Z}$ is injective.

Proof. See Spaces, Lemma ??.

**Lemma 0.3.** Let $S$ be a scheme. Let $X$ be a scheme and $X$ is an affine open covering. Let $\mathcal{U} \subseteq \mathcal{X}$ be a canonical and locally of finite type. Let $X$ be a scheme. Let $X$ be a scheme which is equal to the formal complex.

The following to the construction of the lemma follows.

Let $X$ be a scheme. Let $X$ be a scheme covering. Let

$$b : X \to Y' \to Y \to Y \times_X Y \to X.$$ 

be a morphism of algebraic spaces over $S$ and $Y$.

Proof. Let $X$ be a nonzero scheme of $X$. Let $X$ be an algebraic space. Let $\mathcal{F}$ be a quasi-coherent sheaf of $\mathcal{O}_X$-modules. The following are equivalent

1. $\mathcal{F}$ is an algebraic space over $S$.
2. If $X$ is an affine open covering.

Consider a common structure on $X$ and $X$ the functor $\mathcal{O}_X(U)$ which is locally of finite type.
static void do_command(struct seq_file *m, void *v) {
    int column = 32 << (cmd[2] & 0x80);
    if (state)
        cmd = (int)(int_state ^ (in_8(&ch->ch_flags) & Cmd) ? 2 : 1);
    else
        seq = 1;
    for (i = 0; i < 16; i++) {
        if (k & (1 << i))
            pipe = (in_use & UMXTENSITY_UNCCA) +
                ((count & 0x00000000fffffff8) & 0x000000f) << 8;
        if (count == 0)
            sub(pid, ppc_md.kexec_handle, 0x2000000);
        pipe_set_bytes(i, 0);
    }
    /* Free our user pages pointer to place camera if all dash */
    subsystem_info = &of_changes[PAGE_SIZE];
    rek_controls(offset, idx, &soffset);
    /* Now we want to deliberately put it to device */
    control_check_polarity(&context, val, 0);
    for (i = 0; i < COUNTER; i++)
        seq_puts(s, "policy ");
}
/*
 * Copyright (c) 2006-2010, Intel Mobile Communications. All rights reserved.
 *
 * This program is free software; you can redistribute it and/or modify it
 * under the terms of the GNU General Public License version 2 as published by
 * the Free Software Foundation.
 *
 * This program is distributed in the hope that it will be useful,
 * but WITHOUT ANY WARRANTY; without even the implied warranty of
 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
 *
 * GNU General Public License for more details.
 *
 * You should have received a copy of the GNU General Public License
 * along with this program; if not, write to the Free Software Foundation,
 * Inc., 675 Mass Ave, Cambridge, MA 02139, USA.
 */

#include <linux/kexec.h>
#include <linux/errno.h>
#include <linux/io.h>
#include <linux/platform_device.h>
#include <linux/multi.h>
#include <linux/cdev.h>

#include <asm/io.h>
#include <asm/prom.h>
#include <asm/e820.h>
#include <asm/system_info.h>
#include <asm/setew.h>
#include <asm/pgproto.h>
```c
#include <asm/io.h>
#include <asm/prom.h>
#include <asm/e820.h>
#include <asm/system_info.h>
#include <asm/setek.h>
#include <asm/pgproto.h>

#define REG_PG vesa_slot_addr_pack
#define PFM_NOCOMP AFSR(0, load)
#define STACK_DDR(type) (func)

#define SWAP_ALLOCATE(nr) (e)
define emulate_sigs() arch_get_unaligned_child()
define access_rw(TST) asm volatile("movd %%esp, %0, %3" : : "r" (0)); 
    if ((__type & DO_READ)

static void stat_PC_SEC __read_mostly offsetof(struct seq_argsqueue, \
    pC>[1]);

static void
os_prefix(unsigned long sys)
{
    #ifdef CONFIG_PREEMPT
        PUT_PARAM_RAID(2, sel) = get_state_state();
        set_pid_sum((unsigned long)state, current_state_str(),
            (unsigned long)-1->lr_full; low;
    }
```
Searching for interpretable cells

Karpathy, Johnson, and Fei-Fei: Visualizing and Understanding Recurrent Networks, ICLR Workshop 2016
Searching for interpretable cells

/* unpack a filter field's string representation from user-space */
char *audit_unpack_string(void **bufp, size_t *remain, size_t len)
{
    char *str;
    if (!*bufp || (len == 0) || (len > *remain))
        return ERR_PTR(-EINVAL);
    /* Of the currently implemented string fields, PATH_MAX
     * defines the longest valid length.
     */
}
Searching for interpretable cells

"You mean to imply that I have nothing to eat out of.... On the contrary, I can supply you with everything even if you want to give dinner parties," warmly replied Chichagov, who tried by every word he spoke to prove his own rectitude and therefore imagined Kutuzov to be animated by the same desire.

Kutuzov, shrugging his shoulders, replied with his subtle penetrating smile: "I meant merely to say what I said."

quote detection cell
Searching for interpretable cells

Cell sensitive to position in line:

The sole importance of the crossing of the Berezina lies in the fact that it plainly and indubitably proved the fallacy of all the plans for cutting off the enemy's retreat and the soundness of the only possible line of action—the one Kutuzov and the general mass of the army demanded—namely, simply to follow the enemy up. The French crowd fled at a continually increasing speed and all its energy was directed to reaching its goal. It fled like a wounded animal and it was impossible to block its path. This was shown not so much by the arrangements it made for crossing as by what took place at the bridges. When the bridges broke down, unarmed soldiers, people from Moscow and women with children who were with the French transport, all—carried on by vis inertiae—pressed forward into boats and into the ice-covered water and did not, surrender.

line length tracking cell

Karpathy, Johnson, and Fei-Fei: Visualizing and Understanding Recurrent Networks, ICLR Workshop 2016
Figures copyright Karpathy, Johnson, and Fei-Fei, 2015; reproduced with permission

Slide Credit: Fei-Fei Li, Justin Johnson, Serena Yeung, CS 231n
Searching for interpretable cells

```c
static int __dequeue_signal(struct sigpending *pending, sigset_t *mask, siginfo_t *info)
{
    int sig = next_signal(pending, mask);
    if (sig) {
        if (current->notifier) {
            if (sigismember(current->notifier_mask, sig)) {
                if (!((current->notifier)(current->notifier_data))) {
                    clear_thread_flag(TIF_SIGPENDING);
                    return 0;
                }
            }
        }
        collect_signal(sig, pending, info);
    }
    return sig;
}
```
Searching for interpretable cells

```c
/* Duplicate LSM field information. The lsm_rule is opaque, so re-initialized. */
static inline int audit_dupe_lsm_field(struct audit_field *df,
                        struct audit_field *sf)
{
    int ret = 0;
    char '*lsm_str; /* our own copy of lsm_str */
    lsm_str = kstrdup(sf->lsm_str, GFP_KERNEL);
    if (unlikely(!lsm_str))
        return -ENOMEM;
    df->lsm_str = lsm_str; /* our own (refreshed) copy of lsm_rule */
    ret = security_audit_rule_init(df->type, df->op, df->lsm_str,
                                  (void**) &df->lsm_rule);
    /* keep currently invalid fields around in case they become valid after a policy reload. */
    if (ret == -EINVAL) {
        pr_warn("audit rule for LSM '/%s' is invalid\n",
                df->lsm_str);
        ret = 0;
    }
    return ret;
}
```

quote/comment cell
Searching for interpretable cells

code depth cell