INTERNATIONAL INSTITUTE OF INFORMATION TECHNOLOGY, HYDERABAD

MACHINE LEARNING, QUANTUM COMPUTATION

MACHINE LEARNING AND PHYSICS IN THE VEIN OF OUR PRIMARY WORK

Research Notes

Author: Animesh Sinha

May 5, 2020

Chapter 1

Basic Neural Network Architectures in Keras

1.1 Common Code for Neural Nets

1.1.1 Train models with Saved intermediates

Following is the basic code to run when training the neural Network.

```
model.fit(x=X_train, y=X_train, epochs=25,
validation_data=[X_validation, Y_validation],
callbacks=[keras_utils.ModelSaveCallback(
    model_filename),
          keras_utils.TqdmProgressCallback()],
verbose=0,
initial_epoch=last_finished_epoch or 0)
```

And this is the code to load a presaved checkpoint of the model and start traininf from the last completed epoch. The keras_utils file is available in the snippets, and makes these callbacks available.

```
def load_checkpoint(last_epoch)
    model_filename = 'model.{0:03d}.hdf5'
    last_finished_epoch = None
    if last_epoch is not None:
        s = keras_utils.reset_tf_session()
        last_finished_epoch = 4
        model = keras.models.load_model(
            model_filename.format(last_finished_epoch
            ))
```

Following this, we always save our weights in a file, and then load from it, as follows:

```
encoder.save_weights("encoder.h5")
decoder.save_weights("decoder.h5")
```

1.2 AutoEncoders

1.2.1 Convolutional Autoencoders

Here is the code for the setting up a Convolutional AutoEncoders. Follows a 4 layer Conv-Pool and then 1 Dense layer architecture to encode, then a dense layer followed by Transpose Convolutional layers to decode.

```
import tensorflow as tf
import tensorflow.keras.layers as L

def build_deep_autoencoder(img_shape, code_size):
    """
    Makes a Deep Autoencoder (Encoder-Decoder pair)

    :param img_shape: Shape of the image that is
        being encoded
    :param code_size: Number of the neurons in the
        bottleneck layer
    :returns: The full autoencoder model compiled
        with MSE loss.
    """

    encoder = tf.keras.models.Sequential()
    encoder.add(L.InputLayer(img_shape))
    encoder.add(L.Conv2D(filters=32, kernel_size=(3,
         3), padding='same', activation='elu'))
    encoder.add(L.MaxPooling2D(pool_size=(2, 2),
        padding='valid'))
    encoder.add(L.Conv2D(filters=64, kernel_size=(3,
         3), padding='same', activation='elu'))
    encoder.add(L.MaxPooling2D(pool_size=(2, 2),
        padding='valid'))
    encoder.add(L.Conv2D(filters=128, kernel_size
        =(3, 3), padding='same', activation='elu'))
    encoder.add(L.MaxPooling2D(pool_size=(2, 2),
        padding='valid'))
    encoder.add(L.Conv2D(filters=256, kernel_size
        =(3, 3), padding='same', activation='elu'))
    encoder.add(L.MaxPooling2D(pool_size=(2, 2),
        padding='valid'))
    encoder.add(L.Flatten())
    encoder.add(L.Dense(code_size))

    decoder = tf.keras.models.Sequential()
    decoder.add(L.InputLayer((code_size,)))
    decoder.add(L.Dense((img_shape[0] // 16) * (
        img_shape[1] // 16) ** 256))
    decoder.add(L.Reshape((img_shape[0] // 16,
        img_shape[1] // 16, 256)))
    decoder.add(L.Conv2DTranspose(filters=128,
        kernel_size=(3, 3), strides=2, activation='
        elu', padding='same'))
    decoder.add(L.Conv2DTranspose(filters=64,
        kernel_size=(3, 3), strides=2, activation='
        elu', padding='same'))
```

```
    decoder.add(L.Conv2DTranspose(filters=32,
        kernel_size=(3, 3), strides=2, activation='
        elu', padding='same'))
    decoder.add(L.Conv2DTranspose(filters=3,
        kernel_size=(3, 3), strides=2, activation='
        elu', padding='same'))

    input_image = L.Input(img_shape)
    code = encoder(input_image)
    output_image = decoder(code)
    autoencoder = tf.keras.models.Model(inputs=
        input_image, outputs=output_image)
    autoencoder.compile(optimizer="adamax", loss='
        mse')

    return autoencoder

build_deep_autoencoder(img_shape=(32, 32, 3),
    code_size=100)
```

Chapter 2

Group Theory and Graphs

2.1 Groups and Subgroups

2.1.1 What is a Group?

A group is defined over a Set A and an arbitrary operation $\times$, denoted as: $\langle A | \times \rangle$.

- **Closure:** If a and b are elements in A, then $a \times b$ is also in A.
- **Identity:** There exists e such that $a \times e = a$.
- **Invertability:** There exists a^{-1} such that $a \times a^{-1} = e$.
- **Associativity:** $(a \times b) \times c = a \times (b \times c)$

A Subgroup is a subset of the original group that is itself a group.

One Step Subgroup Test states that if ab^{-1} is in the group H, then H is a subgroup of G.

Two Step Subgroup Test states that if a^{-1} is in H whenever a is in H and ab is in H for all a, b in H, then H is a subgroup of G.

Finite Subgroup Test If H is a non-empty finite subset of a group G, and H is closed under the operation G, then H is a subgroup of G.

Operations that Hold in groups are:

- Uniqueness of Identity (If $x \cdot a = x$ and $x \cdot b = x$, $(\forall x)$, then $a = b = e$)
- Uniqueness of Inverse (If $x \cdot a = e$ and $x \cdot b = e$, $(\exists x)$, then $a = b = x^{-1}$)
- Left and Right Cancellation (If $ab = ac$ then $b = c$. If $ba = ca$ then $b = c$.)
- Socks-Shoes Property ($(ab)^{-1} = b^{-1}a^{-1}$)

2.1.2 Cayley's Table

Cayley's Table is a 2-D matrix of all members of the group a and b on both axis and $a \cdot b$

2.1.3 Subgroups and GCD

Theorem 1.1: Equivalent Cyclic Subpgroups

Let a be an element of order n in a group and let k be a positive integer. Then $\langle a^k \rangle = \langle a^{gcd(n,k)} \rangle$ *and* $|a^k| = n/gcd(n,k)$.

Proof 1.1: Equivalent Cyclic Subpgroups

$(a^{gcd(n,k)})^{\alpha} = a^k$, Since gcd(n,k) divides k. Thereby $\langle a^k \rangle \subseteq \langle a^{gcd(n,k)} \rangle$. Also, $gcd(n,k) = \alpha n + \beta k$, so $a^{gcd(n,k)} = a^{\alpha n + \beta k} = a^{\alpha n} a^{\beta k} = e \cdot a^{\beta k}$, therefore we can state that, $\langle a^{gcdn,k} \rangle \subseteq \langle a^k \rangle$. So we proved that $\langle a^k \rangle = \langle a^{gcd(n,k)} \rangle$.

Next, using the proof in the first part, since the groups are equal their orders are the same, so $|a^k| = |a^{gcd(n,k)}| = \frac{n}{gcd(n,k)}$, since the gcd divides n, it is the least solution to $(a^{gcd(n,k)})^x = a^n$

This has the following crucial corollaries.

- In a finite cyclic group, the order of an element divides the orderof the group.
- Let $|a| = n$. Then $\langle a^i \rangle = \langle a^j \rangle$ if and only if $gcd(n,i) = gcd(n,j)$, and $|a^i| = |a^j|$ if and only if $gcd(n,i) = gcd(n,j)$.
- Let $|a| = n$. Then $\langle a \rangle = \langle a^j \rangle$ if and only if $gcd(n,j) = 1$, and $|a| = |\langle a^j \rangle|$ if and only if $gcd(n,j) = 1$.
- An integer k in Z_n is a generator of Z_n if and only if $gcd(n,k) = 1$.

These facts help us count the number of subgroups in a given set.

2.1.4 Cyclic Groups

Theorem 1.2: Fundamental Theorem of Cyclic Groups

Every subgroup of a cyclic group is cyclic. Moreover, if $|\langle a \rangle| =$ n, then the order of any subgroup of $\langle a \rangle$ is a divisor of n; and, for each positive divisor k of n, the group $\langle a \rangle$ has exactly one subgroup of order k, namely, $\langle a^{n/k} \rangle$.

Theorem 1.3: Number of Elements of Order D

Let G_n be a finite Cyclic Group of order N, if d is a positive Divisor of N, then the number of elements of order D in G of order d is $\phi(d)$.

For any group (i.e. Including non-cyclic), the number of elements of order d is a multiple of $\phi(d)$.

Proof 1.2: Number of Elements of Order D

If d is a divisor of n, then there is only one subgroup of order D of the group G_n. The elements in that group are those which are of the form $\langle a^x \rangle$, such that D and x are coprime. Therefore there can only be exactly $\phi(d)$ elements that are of order d.

If there are no elements of order D in the group, $\phi(d)|0$. Now let $\langle a_1 \rangle$ be a subgroup of order d, it has $\phi(d)$ elements in the subgroup of order d. So on for each $\langle a_i \rangle$ for all i, therefore a multiple of $\phi(d)$.

2.1.5 Abelian Groups

Theorem 1.4: Fundamental Theorem of Abelian Groups

Every finite Abelian group is a direct product of cyclic groups of prime order power. Moreover, the number of terms in the product and the orders fo the cyclic groups are uniquely determined by the group. This is valid **upto isomorphism**.

Proof 1.3: Fundamental Theorem of Abelian Groups

Lemma 1 Let G be a finite Abelian Group of order $p^n m$, where p is a prime that does not divide m. Then $G = H \times K$, where $H = \{x \in G | x^{p^n} = e\}$ and $K = \{x \in G | x^m = e\}$. Moreover $|H| = p^n$

Lemma 2 Let G be an Abelian group of prime-power order and let a be an element of maximum order in G. Then G can be written in the form $\langle a \rangle = K$

Lemma 3 A finite Abelian group of prime-power order is an internal direct product of cyclic groups.

Lemma 4 G be a finite Abelian group of prime-power order. If $G = H_1 \times H_2 \times \ldots \times H_m$ and $G = K_1 \times K_2 \times \ldots \times K_n$, where the H's and K's are nontrivial cyclic subgroups with $|H_1| \geq |H_2| \geq \ldots \geq |H_m|$ and $|K_1| \geq |K_2| \geq \ldots \geq |K_n|$, then $m = n$ and $|H_i| = |K_i|$ $(\forall i)$.

Example 1.1: Using Cardinality and Generators

Prove that every abelian group of order that is a product of primes is also cyclic. Given any abelian Group of order $p \cdot q$. If there is no prime such that

2.2 Special Groups and Their Properties

2.2.1 Permutation Groups

A Dihedral Group (D_n) is a group of all permutations of a n-sided Regular Polygon. The number of elements in this group is 2^n.

Definition 2.1: Permutation Groups

Permutation Groups are a group of permutations, where a permutation is a bijective function from a group to itself. Eg.

$$ExamplePermutation = \begin{bmatrix} A_0 & A_1 & A_2 & A_3 & A_4 & \ldots \\ \alpha(A_0) & \alpha(A_1) & \alpha(A_2) & \alpha(A_3) & \alpha(A_4) & \ldots \end{bmatrix} \tag{2.1}$$

Disjoint Cycle Notation Any permutation can be written as a product of disjoint cycles. Each cyclic subgroup is expressed as a separate disjoint cycle.

$$(1,3)(2,7)(4,5,6)(8) = \begin{bmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \\ 3 & 7 & 1 & 5 & 6 & 4 & 2 & 8 \end{bmatrix} \tag{2.2}$$

These Permutation Cycles can be multiplied together (that is operated by the Group Operator, Function Composition). eg.

$$\begin{aligned} P_1 \times P_2 &= (1,3)(2,7)(4,5,6)(8) \times (1,2,3,7)(6,4,8)(5) \\ &= (1,3)(2,7)(4,5,6)(8)(1,2,3,7)(6,4,8)(5) \\ &= (1,7,3,2)(4,8)(5,6) \end{aligned}$$

We go from Right to Left when multiplying, as function composition applies from Right to left. We see that $1 \to 2 \to 7$, $7 \to 1 \to 3$, $3 \to 7 \to 2$, $2 \to 3 \to 1$ so we get the cycle (1,3,7,2). We continue in this fashion to multiply. By Convention, we can choose not to write down single element cycles.

A Symetric Group (S_n) is a group of all the permutations over the operator Function Composition. There are n! elements in S_n

Every Cycle can be written as a product of Disjoint Cycles Each disjoint cycle is a permutation, and it can always be written as a composition of two or more permutations. Also, any **disjoint cycles can always commute**.

Order of a permutation is defined as the Least Common Multiple of the lengths of the disjoint cycles.

Every Cycle can be Written as a Product of 2 cycles Any permutation is a composition of flips. Also by direct computation, we can prove this. Here is an example: $(01234)(56)(789) = (43)(42)(41)(40)(56)(97)(98)$
The representation of any cycle as a composition of 2 cycles can be classified as even or odd, i.e. any cycle C will require either even number of 2-cycles in all possible breakdowns or odd 2-cycles in all possible breakdowns, but it can never be that some 2-cycle decompositions are odd order and some are even order.

2.2.2 Cosets and Lagrange's Theorem

A Coset of a subgroup H in group G is the set $\{ah \mid H \in a\}$ - Left Coset, or $\{ha \mid H \in a\}$ - Right Coset.

Properties of Cosets of subgroup H in group G are:

1. **a** $\in$ **aH**, since $e \in H$, so $ae \in aH$.

2. **aH = H iff a $\in$ H**, If $ah = H$ then $a = ae \in aH = H$, Also if $a \in H$ then $aH \subset H$ due to closure, $H \subset aH$ as $h = eh = aa^{-1}h = a(a^{-1}h) \in aH$, as a^{-1} is in H (invertability of a in H and then closure).

3. **(ab)H = a(bH) and H(ab) = (Ha)b**, as Associativity holds.

4. **aH = bH iff a $\in$ bH**, as if $aH = bH$ then $a = ae \in aH = bH$. The other way, if $a \in bH$ then $aH = (bh)H = b(hH) = bH$ so they are equal sets.

5. **aH = bH or aH $\cap$ bH = Φ**, Using Property 4, if there exists c in the intersection, then $c \in aH$ and $c \in bH$, so $aH = cH = bH$ thereby $aH = bH$.

6. **aH = bH iff $\mathbf{a^{-1}b} \in$ H**, If there is $c \in aH, bH$ so we can say that $a^{-1}c \in H$ and $b^{-1}c \in H \iff c^{-1}b \in H$ due to invertability. Multiplying both elements in the we get $a^{-1}c \cdot c^{-1}b = a^{-1}b \in H$, again by Closure.

7. **|aH| = |bH|**, We can show a one-one map $aH \to bH$ using the cancellation property, i.e. $ah = bh$ $(\forall h \in H)$.

8. **aH = Ha iff H = aHa^{-1}**, right-multiply both sides by a^{-1}, so $aHa^{-1} = Haa^{-1} = H$.

9. **aH is a subgroup of G, iff a $\in$ H**, aH must have identity e to be a subgroup, since $aH \cap eH \neq \phi \implies aH = eH = H$. By Property 2, $a \in H$. Conversely if $a \in H$ then $aH = H$ is a subgroup.

Theorem 2.1: Lagrange's Theorem

If G is a finite group and H is a subgroup of G, then $|H|$ divides $|G|$. Moreover, the number of distinct (loft/right) cosets of H in G are $|G|/|H|$.

Proof 2.1: Lagrange's Theorem

Using the Properties proven above, $|aH| = |bH| = |eH| = |H|$ and $aH = bH$ or $aH \cap bH = \phi$, therefore $|G|$ is divisible by $|H|$.

The fruitfulness of cosets, when applied to Permutation Groups is displayed here:

Stabilizer of a Point: Let G be a group of permutations of Set S. For each i in S, let $stab_G(i) = \{\phi \in G \mid \phi(i) = i\}$. We call $stab_G(i)$ the stabilizer of i in G.

Orbit of a Point: Let G be a group of permutations of a set S. For each s in S, let $orb_G(s) = \{\phi(s) \mid \phi \in G\}$. The set $orb_G(s)$ is a subset of S called the orbit of s under G.

Eg.: G = {(1),(132)(465)(78),(132)(465),(123)(456),(123)(456)(78),(78)}

Orbits of G	**Stabilizers of G**
$orb_G(1) = \{1,3,2\}$	$stab_G(1) = \{(1),(78)\}$
$orb_G(2) = \{2,1,3\}$	$stab_G(2) = \{(1),(78)\}$
$orb_G(4) = \{4,6,5\}$	$stab_G(4) = \{(1),(78)\}$
$orb_G(7) = \{7,8\}$	$stab_G(7) = \{(1),(132)(465),(123)(456)\}$

Orbit-Stabilizer Theorem states that $|orb_G(i)| \cdot |stab_G(i)| = |G|\ (\forall i)$.

2.2.3 Normal and Factor Subgroups

Normal Subgroups are subgroups H of group G, if $aH = Ha\ (\forall A \in G)$, written as $H \triangleleft G$. i.e., $ah = h'a$, the commutations are fudged a bit, when commuting we are allowed to use a different elements from H.
Normal Subgroup Test: A subgroup H of G is normal in G if and only if $xHx^{-1} \subseteq H\ (\forall x \in G)$.

Factor Groups (or Quotient Groups) are the groups, such that H is a normal Subgroup of G, $G/H = \langle \{aH \mid a \in G\} \mid\mid (aH)(bH) = abH \rangle$. This is because the normal subgroups are such that their left or right cosets are themselves subgroups. The operation is the composition of the operation that generated the cosets.

Internal Direct Product: We say that G is the internal direct product of H and K and write $G = H \times K$ if H and K are normal subgroups of G, $G = HK$, $H \cap K = \{e\}$

2.3 Isomorphism and Homomorphism

2.3.1 Isomorphisms

Definition 3.1: Isomorphism

A Isomorphism is a bijective mapping from one group onto another wherein if $a \times b = c$ then $\Phi(a) \times \Phi(b) = \Phi(c)$. That is the mapping preserve the result of every operation, and every inverse.

Automorphism is an Isomorphism that exists from a group onto itself.

Example 3.1: Disproving Isomorphisms

The Set of Real numbers can never be isomorphic to proper subset of itself under the operation addition.

The homomorphisms of real numbers to its proper subsets under addition are highly constrained: $\phi(a) + \phi(a) = \phi(2a) \implies \phi(m) = qm$. Therefore there can only be a multiplicative map that holds the homomorphism.

Any subgroup of rational numbers can be generated by a subset of the series if prime inverses: $\{\frac{1}{2}, \frac{1}{3}, \frac{1}{5}, \frac{1}{7}, \frac{1}{11}, \frac{1}{13}, ...\}$, the set of all prime denominators. Since both of our subgroups are generated by a subset of this set, there must exist a map, just a multiplicative factor, ϕ, from $A \to B$. This never exists, since there is no such factor that maps the set of prime numbers to another set of prime numbers (or the reciprocals thereof).

Example 3.2: Proving Isomorphisms

From $\langle \mathbb{R} | + \rangle$ to $\langle \mathbb{R} | \times \rangle$, show an isomorphism exists.

We have $\phi(x) = 2^x$, the mapping is bijective, and $\phi(x) \times \phi(y) = 2^x \times 2^y = 2^{x+y} = \phi(x+y)$

Cayley's Theorem states that every group is isomorphic to a group of permutations.

2.3.2 Homomorphism

Definition 3.2: Group Homomorphism

Homomorphism ϕ is a mapping from a group G to $\bar{G}$ is a mapping that preserves the group operation, i.e. $\phi(ab) = \phi(a)\phi(b)$ $(\forall\ a, b \in G)$.
(The mapping may not be bijective, as opposed to Isomorphism)

Definition 3.3: Kernel of a Homomorphism

he kernel of a homomorphism ϕ from a group G to a group with identity e is the set $\{x \in G \mid \phi(x) = e\}$. The kernel of ϕ is denoted by $Ker\phi$.
(All elements in G that map to the identity, for isomorphism it's just the identity element - trivial subgroup)

2.4 Rings and Fields

2.4.1 Rings

A Ring is a set with 2 binary operations on it such that the following properties hold true. (Z is a arbitrary set, and + and × are arbitrary operations in $\langle Z|+,\times\rangle$, also + is the first and × is second operation on the ring).

- Associativity over +: (a + b) + c = a + (b + c)
- Commutativity over +: a + b = b + a
- Identity over +: a + 0 = a must exist for some 0
- Invertable over +: a + (-a) = 0 must exist for some -a
- Associativity over ×: $a \times (b \times c) = (a \times b) \times c$.
- Distributivity of × over +: $a \times (b + c) = (b + c) \times a = (a \times b) + (a \times c)$.

In summary, A Ring is a Abelian Group over +, and Associative and Distributive over ×.

A subring S of a ring R is the same operations defined over a subset of elements such that they in themselves form a ring.

Subring Test A non-empty subset of a ring is a subring if it is closed unber subtraction $(a-b)$ and multiplication $(a \times b)$.

2.4.2 The Integer Domain

Rings were invented to abstract the algebraic properties of Integers. However they lose essential features in this abstraction, those of **Existance of Unity, Commutativity, and Cancellation**

Zero Divisors are elements of a Commutative Ring such that there is a non-zero element $b \in R$ with $ab = 0$.

Integral Domain is a commutative ring with unity and no zero-divisors. (It may also be defined as a Commutative Ring with Cancellation, Equivalent)

2.4.3 Fields

A Field is a commutative ring with unity in which every nonzero element is a unit (i.e. Invertable).
Every finite integral domain is a field. (Finite Order: So every element is invertible)

Characteristic of a Ring is the least positive integer n such that nx = 0 for all x in the Ring. If no such integer exists then it is 0.
If R is a ring with unity (1), then the characteristic of n is the order of 1, unless its ∞ in which case characteristic is 0.
Characteristic of an integral domain can only be 0 or prime. Its because if the field is finite, then $0 = n \cdot 1 = (p \cdot 1)(q \cdot 1)$. So, either $p \cdot 1 = 0$, or $q \cdot 1 = 0$. Since p or q are smaller than n, then p or q is the characteristic, not n.

Chapter 3

Particle Physics for Data Scientists

3.1 The Preliminaries

3.1.1 Problems with the standard model

What makes us unhappy?

- Matter and Antimatter inequivalence.
- 19 Arbitrary constants.
- Why is Gravity so weak.

3.1.2 The Particles we want to detect

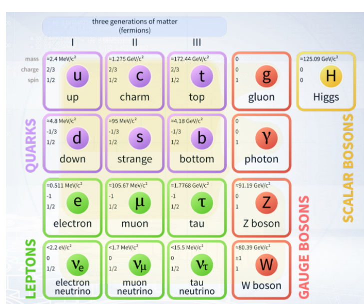


Figure 3.1: List of Particles of different types

The following are types of particles we want to defect.

- muon
- pion
- electron
- kon
- proton

3.1.3 The Experiments in LHC

There are 4 major detectors

- ALICE (A Large Ion Collider Experiment) is a heavy-ion detector on the Large Hadron Collider (LHC) ring. It is designed to study the physics of strongly interacting matter at extreme energy densities, where a phase of matter called quark-gluon plasma forms.

- ATLAS (A Toroidal LHC ApparatuS) is one of two general-purpose detectors at the Large Hadron Collider (LHC). It investigates a wide range of physics, from the search for the Higgs boson to extra dimensions and particles that could make up dark matter. Although it has the same scientific goals as the CMS experiment, it uses different technical solutions and a different magnet-system design. It has a cylindrical structure and measures particles in all directions.

- CMS (Compact Muon Solenoid) is a general-purpose detector at the Large Hadron Collider (LHC). It has a broad physics programme ranging from studying the Standard Model (including the Higgs boson) to searching for extra dimensions and particles that could make up dark matter. Although it has the same scientific goals as the ATLAS experiment, it uses different technical solutions and a different magnet-system design.

- LHCb (Large Hadron Collider Beauty) experiment specializes in investigating the slight differences between matter and antimatter by studying a type of particle called the "beauty quark", or "b quark". It is a single arm forward spectrometer.

We smash bunches of protons ('events'), record the pixels ('hits'), reconstruct trajectories ('jets', 'showers', 'tracks'), and we perform Statistical analysis on them.

A Trigger System is a system that uses criteria to rapidly decide which events in a particle detector to keep when only a small fraction of the total can be recorded.

3.1.4 Simulation Package

- http://www.genie-mc.org/

- http://home.thep.lu.se/Pythia/: Nutrino Simulations

- GEANT4: http://geant4.web.cern.ch/: Particles interacting with matter.

- FLUKA: http://www.fluka.org/fluka.php

3.1.5 Feynman Diagrams

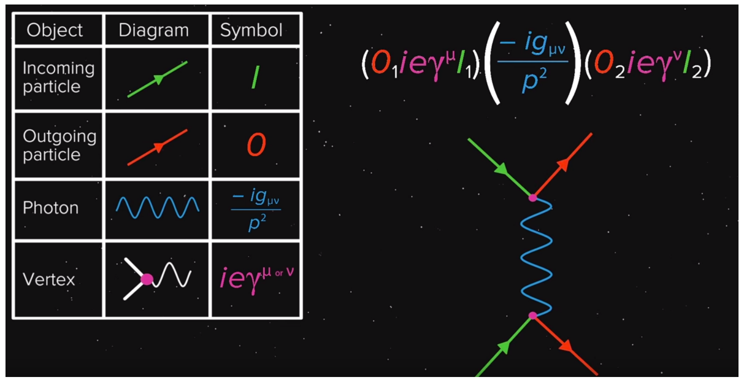
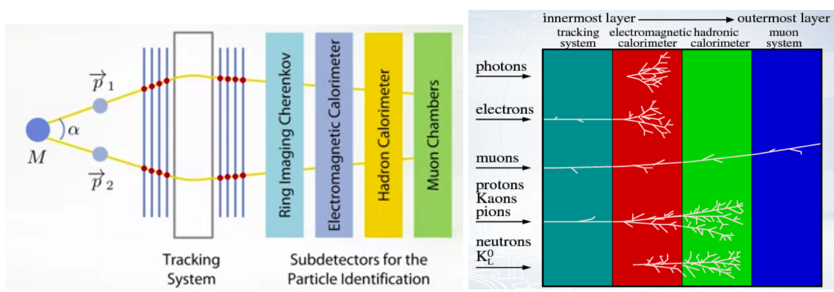


Figure 3.2: A Sample Feynman Diagram for scattering of Two electron by the transfer of one Photon

3.2 The Large Hadron Collider Setup



(a) Detector Setup of the LHC (b) The tracks made in the Chambers

3.2.1 Tracking System

The first system of detectors, stands before particle collision area. Important for parameter estimation.

There are several layers of sensors that measure hits, and allow us to recognise particle trajectories. There is also a Magnetic Field that allows us to measure momentum (using radius of curvature in the field).

We have the following conservation equationsm wgeb a particle D^0 with mass m breaks into a K^- with mass m_1 and a π^+ with mass m_2, and they go away from each other at angle α:

$$
\begin{aligned}
E_m &= E_1 + E_2 && (3.1)\\
\hat{p_m} &= \hat{p_1} + \hat{p_2} && (3.2)\\
E^2 &= p^2c^2 + m^2c^4 && (3.3)\\
M^2 &= m_1^2 + m_2^2 + \frac{2}{c^4}(E_1E_2 - p_1p_2c^2 cos\alpha) && (3.4)
\end{aligned}
$$

Problem: Track Pattern Recognition

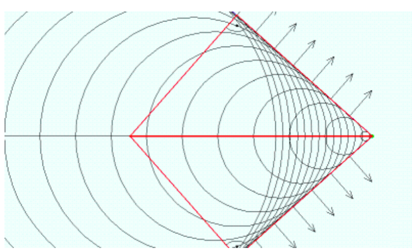
Recognizing hits that belong to the same track. Currently we have the following methods.

- Half Transform and Kalman Filtering. (Statistical, computationally cheaper.)
- Hopfield Neural Netorks. (Denby Peterson and Cellular Automaton.)
- Convolutional Nerual Networks (classify result as correct or wrong). Recurrent Neural Netorks (predict the next hit location).

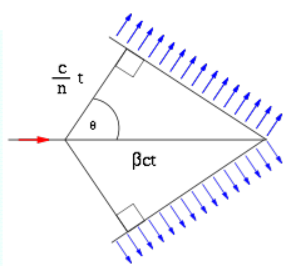
We can then combine these particle tracks into decays.

3.2.2 Ring Imaging Chernekov Detector (RICH)

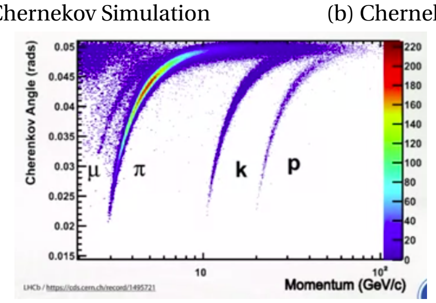
This is the first detector because it does not affect the flight of the particle.



(a) Chernekov Simulation



(b) Chernekov Angles



(c) Chernekov Radiation Graphs

Figure 3.4: Registers in Processor Design

Here is the angle of the Chernokov radiation, derivable using simple geometric means and some special relativity in terms of the momentum.

$$p = \frac{mc\beta}{\sqrt{1 - v^2/c^2}} \tag{3.5}$$

$$cos\theta = \frac{1}{n\beta} = \frac{\sqrt{p^2 + m^2c^2}}{np} \tag{3.6}$$

From the figure above, and the equation for it's analytical feel, that we can figure out the mass of the particle and thereby which particle it is.

3.2.3 Calorimeter - Electromagnetic and Hadronic

Electromagnetic stops all but than muons and quarks. Hadron calorimeter stops the quarks.

$$E_C = E_0 e^{\frac{x}{X_0}} \tag{3.7}$$

$$X_{max} = X_0 ln(\frac{E_0}{E_c}) \tag{3.8}$$

$$N = E_0/E_c \tag{3.9}$$

After this we have crystals and scintillation counters, that measure the count and energy of particles, The electrons and photons, and since we measure energy, we already have momenta from the tracking system., this gives us the Mass, so we can identify them.

3.2.4 Muon Chambers

Muons do not interact well with matter. So we have multiple layers of metal to slow down the muons, they activate the chambers, we will extrapolate the data to get which particles in the tracking system were muons. We can also get the energy from how far they go in the muon chambers.

3.2.5 How to make a classifier uniform

The problem

$$1 - false\, positive\, rate = background\, rejection \tag{3.10}$$

$$true\, positive\, rate = signal\, efficiency \tag{3.11}$$

We want to have no dependence of the ROC AUC on all the momenta.

$$L_{ada} = \Sigma e^{-\gamma_i s_i} \tag{3.12}$$

$$L_{flat} = \Sigma_b w_b \int |F_b(s) - F(s)|^2 ds \tag{3.13}$$

$$L_{adaflat} = L_{ada} + \alpha L_{flat} \tag{3.14}$$

This is imposing an additional loss over the ADA boost function to make it less dependent on the momentum, alpha is a parameter we can tune to weight flatness over fit quality.

3.2.6 Adversarial Neural Networks

Minimizing dependencies on Mass, Momentum, etc. can also be done using adversarial neural networks. One network performs classification, and feeds it's output to the other which tries to guess the momentum/mass from the output, if it can do this well, the model is bad, i.e. not flat. So, our new loss is $Loss_{classification} + f(Loss_{adversarial})$.

3.3 Discovering New Physics

3.3.1 NoEther's theorem

NoEther's theorem claims that the conservation laws we have are based on certain symetries. Here is a list of some of them.

- Time Inversion: Energy
- Space Translation: Linear momentum
- Space Rotation: Angular momentum
- Charge Conjugation and Parity: Time Isotropy
- Charge, Lepton Number: Gauge Symetries

3.3.2

CvM test

$$CvM = \Sigma_{region} \int |F_{region}(s) - F_{global}(s)|^2 dF_{global}(S) \tag{3.15}$$

3.3.3 Doping